Synthetic Brain Cancer Image Generation and Classification: A Performance Analysis

Francesco D'Aprile, Anthony Di Pietro, Sara Lazzaroni, Tommaso Mattei

1 Abstract

Healthcare is a very complex and important topic; computer science, like any other science, aims to improve the lives of humans. Artificial intelligence is a tool that recently is helping the work of experts and physicians, and it could be the key to achieving new unseen milestones.

However, artificial intelligence is a data-hungry approach that requires possibly sensible data to solve healthcare problems. Training machine learning models on sensible data may not be ethically correct.

Our motivation is to solve the ethical, privacy and sensible problems that lie behind the use of healthcare data. The key that might be able to achieve that is the generation of new, synthetic data.

2 Introduction

Learning models on sensible data, especially healthcare, may not be ethically correct. The solution might lie in creating a new, synthetic dataset that is very similar to the real dataset. Our work focused on the creation of synthetic data and its likelihood with respect to the real, original data.

Our approach to create the synthetic dataset leverages on two models: Variational Autoencoders (VAE) and Generative Adversarial Network (GAN). The data was evaluated with two classification models: Convolutional Neural Network (CNN) and Variational Autoencoders (VAE).

We want to compare how the classifications models perform when trained on different kind of datasets (real, synthetic, mixed).

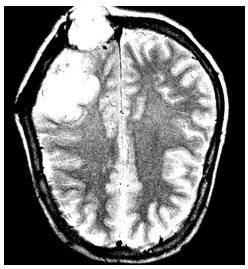
3 Related Work

The research has been really flourishing about computer science being applied to healthcare problems. Some notable work that is related to our project is CovidGAN [1] that focused on data augmentation to classify Covid-19 and F&BGAN to Improved Lung Nodules Classification [2] that generated synthetic data to improve lung classification.

Changhee Han [3] shows that applying two-step GANs to detect a tumor in the BRATS dataset has boosted the sensitivity of the model. The same dataset was used in another study in the same year (2019) by the same main author along with other authors [4] in another study has used the Conditional Progressive Growing of GANs (CPGGANs) model which improved the accuracy, yet the test accuracy decreases, because the classifier recognizes the synthetic images.

4 Dataset

The dataset that was used for this project is Br35H Dataset [5] that contained a total of 3000 examples (1500 negatives and 1500 positives) of MRI brain scans. During our research, we noticed that the dataset contained data-augmented points and the brains were detected with different techniques (t1, t1ce, t2, flair) and different sizes (between different images and between their own axes), therefore different resolutions and aspect ratios. Furthermore, not all the images have the same background.



(a) 'Flair' Brain MRI of size (587×630)

(b) 'T2' Brain MRI of size (197×256)

Fig. 1: Two images of the dataset, which have a stark difference due to different scanning methods

All the aforementioned reasons represent a challenge for the models to actually learn the features needed for both the generation and the classification.

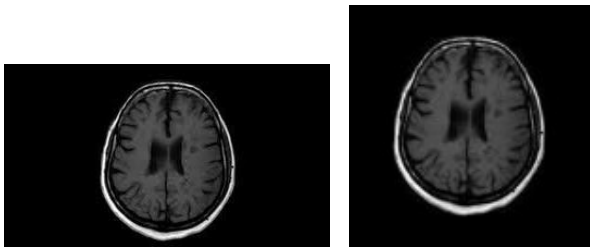
5 Proposed Method

Our project focused on two main tasks: generation and classification.

5.1 Data Generation

Pre-processing - To achieve a reasonable data generation, we focused on the original data preprocessing. To maintain the aspect ratio and other

qualities about the data, we needed to individuate the bounding box where the brain was contained and transfer it to a background that will be shared across all the data. In this way, we were able to make our dataset of the same size and of the same aspect-ratio. We also normalized the pixels with a range $[0, 1]$. An example of an image after the pre-processed step is shown at Figure 2.



(a) Original image (b) Pre-processed image

Fig. 2: Comparison in pre-processing step

Models - In our approach, we implemented two distinct generative models: Variational Autoencoder (VAE) and Generative Adversarial Network (GAN).

Generative Adversial Network. Gans consist into two neural networks, a generator and a discriminator, trained together in a competitive way. The generator attempts to create the most realistic images possible. Instead, the discriminator attempts to distinguish between real or fake images. The adversarial process drives the generator to create increasingly realistic images. Gan is hard to train because the training process involves a balance between the generator and the discriminator. Additionally, Gan requires significant computational resources to achieve optimal result.
Our model is composed by:

- The generator takes as input a latent vector of size 100 and progressively transforms it into an image of size (128, 128). It begins with a fully connected layer that maps the latent vector to a feature map of size (128, 8, 8). This reshaping is followed by several convolutional layers aimed at upsampling the feature maps to higher resolutions. The generator uses batch normalization and ReLU activations after each convolutional layer.
- The discriminator takes as input an image of size (128, 128) and classifies it as either real or fake. It consists of four convolutional layers, each followed by LeakyReLU activations. After these convolutional layers, the output is flattened and passed through a fully connected layer to produce a scalar value, representing the probability of the image being real.

Variational Autoencoder. VAEs consist into two components, an encoder and a decoder. The encoder learns to encode input data into a lower-dimensional latent space. It returns the parameters of a Gaussian distribution, from which a sample latent variable (denoted as z) is chosen. The decoder uses z to generate a new image. The goal of training process is minimizing the loss function expressed through two factors: reconstruction loss, the difference between the input data and the reconstructed data to ensure the generated image is as close as possible to the original ones; KL divergence loss, the difference between the learned latent distribution and a prior distribution. Generally, the VAE is easier to train compared to others generative models. So this approach gives us better results.
Our model is composed by:

- The encoder takes as input an image of size (128, 128). It consists of six convolutional layers, each followed by batch normalization and LeakyReLU activations to help the model learn complex patterns. The layers progressively reduce the spatial dimensions of the input while increasing the depth of feature maps. The encoder ends with two fully connected layers that produce the mean and log-variance parameters of the Gaussian distribution in the latent space. The latent dimension is set to 512.
- The decoder essentially reverses the process of the encoder, progressively increasing the spatial dimensions of the feature maps to match the input image size. Starting from a fully connected layer that reshapes the latent vector into a (2,2) feature map with 1024 channels, the decoder upsamples the data back to the original image size (128,128) using convolutions. Each upsampling step is followed by batch normalization and ReLU activations to help generate realistic images. The final output layer uses a sigmoid activation function to ensure that the reconstructed image has pixel values in the range [0, 1].

Generation - Given the dichotomy of the problem, where images either contained or not contained a tumor, we needed to train two models for each of the two architectures: Varietional Autoencoder (VAE) and Generative Adversarial Network (GAN). For each architecture, we trained two distinct models to specialize in the generation of positive and negative examples. One model focused exclusively on the positive examples (brains that contained a tumor), to be able to generate additional positive examples. Dually, the second model was trained solely on the negative examples (brains without tumor) to generate negative samples. This process was applied for both the VAE and GAN architectures.
Since our goal is to compare the performance of classification models trained also on mixed dataset or entirely synthetic one, we generated 1500 positive examples and 1500 negative examples with both VAE and GAN models, achieving a total of 3000 positive and negative examples. This approach ensures

a balance representation consistent with the original dataset. In this way, we maintain the balance representation of the original dataset.

5.2 Classification

For the classification task, we focused on two models: CNN and VAE.

CNN architecture - The BrainCNN model is a convolutional neural network designed for tumor classification, featuring a series of convolutional and fully connected layers. It begins with three convolutional blocks, each consisting of a convolutional layer, batch normalization, ReLU activation, and max-pooling to progressively extract spatial features and reduce dimensionality. The first block uses 32 filters, the second 64, and the third 128, all with a 3×3 kernel size and a stride of 1. Following the convolutional layers, the output is flattened and passed through a fully connected network with three hidden layers of sizes 1024, 512, and 128 neurons, each activated by ReLU and regularized with dropout layers at a configurable probability (set as 0.5). The final layer outputs logits for two classes, completing the tumor classification pipeline.

VAE architecture - The BrainCVAE is a convolutional variational autoencoder (CVAE) designed for generating and reconstructing images from a learned latent space. The model comprises an encoder, a decoder, and a reparameterization mechanism for sampling latent vectors. The encoder extracts features using two convolutional layers with ReLU activation, followed by a fully connected layer that produces the mean and log-variance of the latent distribution. Using the reparameterization technique, latent vectors are sampled and passed to the decoder. The decoder reconstructs the images through a fully connected layer that reshapes the latent vectors, followed by two transposed convolutional layers with progressively larger kernel sizes. A custom loss function combines the reconstruction loss and the KL divergence to ensure efficient latent space learning.

VAE classification - VAE does not perform the classification task natively, it was necessary to make some additional operations to make it work. The idea is that, given the trained model exclusively on healthy examples, the corresponding latent space does not retain the information about the tumorous regions in the brain. This means that the reconstructed images diverge more when attempting to reconstruct a tumorous input example compared to a non-tumorous one. We introduce the concept of *reconstruction error* which is defined as the pixel-wise absolute difference between the original image and the reconstructed image given the *healthy* latent space.

This idea introduced different classification techniques:

1) **Global image absolute error magnitude**: this is the first method used, which observed the reconstruction error. This technique leverages on two hyperparameters, expressed as percentages, "'anomaly_threshold"' and "'tumor_index"': the former focuses on how divergent the reconstruction error (pixel-wise) must be to be classified as an anomaly, while the latter ensures a minimum set of anomalies across all the reconstruction error to classify the example as tumorous/non-tumorous.
2) **Application of DBSCAN [6] on the reconstruction error** We leveraged on the Scikit-Learn implementation of DBSCAN to analyze the denser cluster, which supposedly contains the tumorous region. The example was deemed as positive if such biggest cluster contained more than "'tumor_index"' points, a hyperparameter expressed in percentage across the total image area. This technique may be able to accomplish a much more complex task which is unsupervised tumor detection.
3) **Singular-example KMeans [7]** We leveraged on the Scikit-Learn implementation of KMeans to distinguish the tumorous cluster and the non-tumorous cluster by assigning two centroids. To actually classify, we used the "'tumor_index"' hyperparameter that models how big a cluster should be. Such as the previous point, this techinique could be able to attempt unsupervised tumor detection.
4) **Global KMeans** Using the same implementation as the previous point, instead of just looking at a single image, we looked at the space spanned by all the reconstruction errors, which is a $N \times (H \cdot W)$ matrix where N is the number of input images and $H \cdot W$ represents their size. We leverage on two hyperparameters. "'anomaly_threshold"' which transforms the input matrix that contains real values to a boolean matrix, populated with 0 and 1. The idea is that where the reconstruction error diverges more, there is an anomaly which, in our case, should represent the tumor region. The other parameter used is "'alpha"' that restricts the clusters that, empirically, were too big.

Training and Validation - The training split was different for the CNN and VAE.

The CNN dataset split - The first model that we used to solve the classification task was the CNN. It was trained on a randomly split dataset into the three standard sets: 70% training, 20% validation and 10% test.

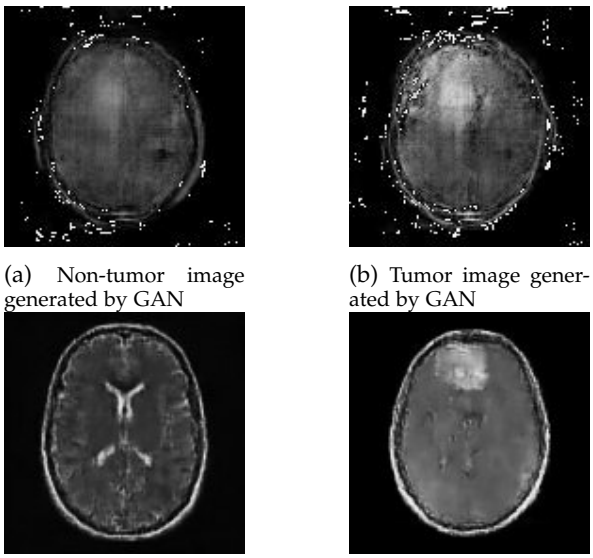
The VAE dataset split - The second model used for classification is the Variational Autoencoder. It was trained exclusively on positive examples which amounted to 75% of the total amount of positive ex-

amples. The idea is to detect the most divergent examples (tumorous) given a latent space that learned exclusively on healthy examples. The validation set contained both positive (40% of the original dataset) and negative examples (10% of the original dataset). The test set also had a similar structure and contained both negative (15% of the original dataset) and positive (60% of the original dataset) examples.

6 Experimental Results

In this section we illustrate the results for both the generative and the classification part. The classification focused on the metrics accuracy, precision, recall and F1 score with particular attention to recall that, when faced with healthcare problems, is very important.

Generative task - The GAN model struggles to generate tumors on the positive examples. Also, the synthetic GAN-generated data contained too much noise and the model wasn't able to actually replicate the complex anatomical complexity of the human brain, this probably was the result of a too-small dataset fed to a very data-hungry model. For this reason, the generated examples of the GAN model were discarded and there were not computed relevant metrics for them.
On the other hand, the simpler VAE architecture was able to generate reasonable positive and negative samples and were more realistic. However, the model may struggle to actually create tumorous area for the positive examples.
Examples of synthetic images are shown at Figure 3.



(a) Non-tumor image generated by GAN

(b) Tumor image generated by GAN

(c) Non-tumor image generated by VAE

(d) Tumor image generated by VAE

Fig. 3: Synthetic images

Classification task - The classification models CNN and VAE behaved very differently. CNN achieved almost perfect results after only 20 epochs. VAE overall, with all the illustrated techiniques, performed worse on the classification task, this may be because the dataset images are not IID (Independently, Identically Distributed) due to different scanning techiniques.

For the global image absolute error magnitude approach, the results showed that the performance was decent but not outstanding. The model is able to detect the anomalies well, but unfortunately many points that shouldn't be tumorous are detected. An example of a positive and negative example and their reconstruction is shown at Figure 4 and Figure 5 respectively.

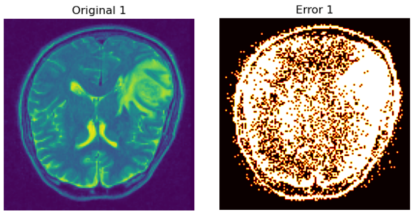


Fig. 4: Positive example and its reconstruction error

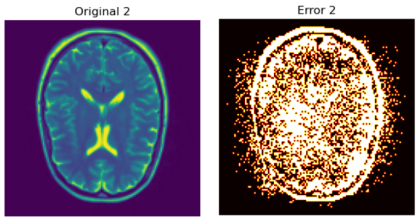


Fig. 5: Negatives example and its reconstruction error

DBSCAN seems to extract meaningful information and it should be able to perform an actual unsupervised tumor detection, however the computational complexity of the implementation and the huge domain of its intrinsic hyperparameters rendered it unfeasible to actually obtain good results of out it.

Singular-example KMeans, unfortunately, was not able to actually detect the tumorous regions as clusters, even after we encoded the spatial locality as features.

Global KMeans, instead, was able to improve the results of global image absolute error magnitude. This shows us that KMeans actually performs better when it looks at the whole input set at once, rather to compute image-per-image.

The Table 1 shows the results of all the applied techiniques and models.

Summarizing the results - Evidently, CNN works extremely well with the plain, real dataset, achieving an almost perfect score across all the metrics. However the CNN wasn't able to learn really well the features given the synthetic dataset. When faced with a mixed dataset that contained both real and synthetic samples, the model seems to classify each image as positive.

Model	Train Data	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
CNN	Real	96.67	95.42	97.99	96.69
CNN	Synthetic	58.16	57.53	56.94	57.24
CNN	Mixed	49.97	49.98	99.93	66.64
VAE (GIAEM)	Real	60.44	60.95	67.11	56.76
VAE (GIAEM)	Synthetic	80.08	68.26	66.22	67.09
VAE (GIAEM)	Mixed	81.24	69.99	64.11	65.95
VAE (Global KMeans)	Real	82.93	73.55	67.58	69.66
VAE (Global KMeans)	Synthetic	79.39	68.26	69.44	68.80
VAE (Global KMeans)	Mixed	78.8	39.88	49.27	44.08

TABLE 1: Comparison of models performance with different training, test and validation sets and techniques. GIAEM stands for Global Image Absolute Error Magnitude. Validation and test are always performed on real data.

The results of the VAE model, on the other hand, are more complex. The metrics on the real data were not the best. It achieved decent results, but far from the CNN results. The synthetic data, generally, improve or retains the model's performance.

7 Conclusions and Future Work

The Br35H [5] dataset was not the best dataset to work on with our task because the data were not Independent and Identically Distributed (IID). The MRI brain scans were obtained with different techniques, also data augmentation may make the generative model struggle more.

The idea would be to find a better dataset for our task; this could be the key to make the generative models create more realistic examples. Also, having a bigger dataset could make us able to achieve meaningful results with GAN.

The VAE emerged as the best-performing generative model, producing images that were less noisy compared to those generated by the GAN. Furthermore, the VAE demonstrated superior accuracy in capturing skull size and difference between the generated tumorous scans from non-tumorous ones, however it seems that the model, in some positive examples, was not able to actually generate the tumorous region, which influences negatively the results of the classification methods.

The CNN proved to be highly effective for the classification task, achieving near-perfect results. While the VAE consistently reconstructed input images from its latent space, it showed lower discrimination for positive examples, likely influenced by the dataset's characteristics. Tumor detection using DBSCAN showed potential, but the extensive hyperparameter tuning required rendered it impractical. On the other hand, KMeans struggled with consistent object detection. However, it demonstrated an ability to identify tumorous examples by analyzing the space of the errors (absolute difference between the original image and its reconstruction).

8 Roles

We split into two teams: one for the classification task and one for the generation one.

- Francesco D'Aprile and Sara Lazzaroni worked on the generative models and the pre-processing
- Anthony Di Pietro and Tommaso Mattei focused on the classification task.

Everything else was done by the whole group together.

References

[1] CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection Abdul Waheed et al. https://arxiv.org/pdf/2103.05094

[2] Synthetic Medical Images Using F&BGAN for Improved Lung Nodules Classification by Multi-Scale VGG16 Defang Zhao et al https://www.mdpi.com/2073-8994/10/10/519

[3] Combining Noise-to-Image and Image-to-Image GANs: Brain MR Image Augmentation for Tumor Detection, available at https://ieeexplore.ieee.org/document/8869751

[4] Infinite Brain MR Images: PGGAN-based Data Augmentation for Tumor Detection, available at https://arxiv.org/abs/1903.12564

[5] Br35H :: Brain Tumor Detection 2020 https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection

[6] DBSCAN, available at https://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD-96.final.frame.pdf

[7] KMeans, available at http://www.stat.yale.edu/~pollard/Papers/Pollard82ITT.pdf