Classification of Brain Tumors from MRI Images Using a Capsule Classifier and Image Synthesis

Tel Aviv University. DLMI course (0553-5542). July 2020.

Asaf Zorea Zrien
ID xxxx
zoreasaf@gmail.com

Marom Dadon
ID xxxx
drunsnv@gmail.com

***Abstract*—The biggest problem for classifying magnetic resonance images (MRI) with deep learning techniques lies in the number of labelled data. Although recent works in the field of neural network have shown promising abilities of classifying brain tumors from brain MRI images, they used very deep and complex network architectures. Simpler network architecture requires fewer resources, and enable to use real time applications on mobile platforms. To this end, we introduce a new deep learning method to classify brain tumor of three tumor types. Our solution combines synthesise of additional data using a conditional generative model, with a new convolutional capsule architecture for the classification.**

We evaluated the performance both qualitatively and quantitatively. Qualitatively, the conditional GAN was capable at synthesising images of different appearance, for the same underlying skull geometry. Moreover, the features learned by the conditional GAN are often semantically meaningful groups, covering structures such as skull and tumor. The classificaiton performance was evaluated quantitatively using accuracy and F1 measures. The best results of 93% accuracy and F1 score of 92%, were obtained when the classifier pre-trained on 7,000 synthesised images and then trained on the original data using a 7-fold cross-validation. Our method performs as well as the Resnet50 state-of-the-art deep network, with 9x less parameters.

***Index Terms*—Capsule Network, Generative Adversarial Network, Brain Tumor, Classification, Magnetic Resonance Images, Image Synthesis, Image-to-Image Translation**

I. Introduction

A. Biological Background

Brain and other nervous system cancer is the 10th leading cause of death for men and women. According to the American Society of Clinical Oncology (ASCO), it is estimated that in this year 23,890 adults in the United States will be diagnosed with primary cancerous tumors of the brain and spinal cord, and 18,020 adults will die [1]. In general, there are two types of brain tumors, benign and malignant. Benign tumors differ from malign tumors in that benign generally do not spread to other organs and tissues and can be surgically removed. Brain tumors can be divided into classes based on the affected area and shape, some are Glioma, Meningioma, and Pituitary. Glioma is a brain tumor that arise from brain tissues other than nerve cells and blood vessels. On the other hand, Meningioma arise from the membranes that cover the brain and surround the central nervous system, whereas Pituitary tumor is locate inside the skull. The reason that we want to classify between the brain tumor classes, lies in the fact that Meningioma tumors are typically benign and Glioma tumors are commonly malignant. Unlike Glioma tumors, Pituitary tumors, even if they are benign, can cause other level of medical damage because they are fast-growing tumors. Because of the information mentioned above, the classification between these type of brain tumors represents different clinical diagnostic and may lead to different kind of medical treatment. Early and precise diagnostic is important because earlier studies have showed that when cancer is found early, it may be easier to treat and reduce the chance of dying from the disease. The affected brain areas of the different types of tumors, are shown in Figure 1.

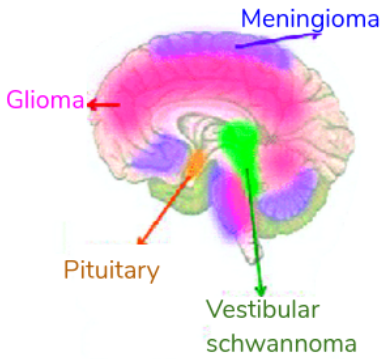

Fig. 1. Affected brain areas of the different types of tumors: Glioma, Pituitary and Meningioma. Vestibular schwannoma is another type of brain tumour that is not shown in our dataset.

The most common method for diagnose brain tumors is magnetic resonance imaging (MRI). The result from an MRI scan are interpreted by expert radiologists. However, analysis of large number of MRI scans is difficult and done by human observation in a subjective manner. Errors and discrepancies in radiology practice are uncomfortably common and early brain tumor detection mostly depends on the experience of the radiologists [2]. Recent works in the field of neural network have shown promising abilities of early detection of various illnesses and other medical conditions that requires medical care. Different machine and deep learning methods for image segmentation and classification are applied in MRI image processing to provide radiologists with a second opinion.

B. Summary of Our Contributions

The biggest problem of classifying brain tumors from MRI images is lack of labelled brain tumor MRI images. In order to allow the use of simpler network architecture, that requires fewer resources, and enable to use real time and mobile platforms, we proposed a method that combines synthesise of additional data using a conditional generative model, with a new convolutional capsule architecture for classification.

We based on the pix2pix architecture for the cGAN [8]. We changed the input mask to 1D channel (instead of 3D channel) and modified the network so it could handle image size of (512x512) instead of (256x256). We also made the network compatible to depth of 16-bit. Furthermore, we added a latent vector component that was processed through a fully connected layer and then concatenated with the input mask image. We intended to introduce randomness through latent vector and to increase the variation of the synthesised images. The conditional generative model was capable at synthesising images of different appearance, for the same underlying skull geometry, while reshaping the tumor label using rotate, shear and scale transformations. However, the network wasn't able to produce varied images with the same skull-tumor geometry with different latent vectors. Following the idea that capsule networks were made to better encode spatial relationships between features than standard CNNs, we proposed a new convolutional capsule network to classify the brain tumors. The best result of 93% accuracy and F1 score of 92%, was obtained when the classifier pre-trained on 7,000 synthesised images and then trained on the original data using a 7-fold cross-validation. Our method performs as well as Resnet50 state-of-the-art deep network, with 9x less parameters.

II. Related Work

A. Caspule Networks

Capsule networks were first presented by Sabour et al at 2017 [5]. Capsules are a vector that are specifying the features of the object and its likelihood. These features can be any of the parameters such as "pose" (position, size, orientation), hue, texture, etc. The output of the capsule is a vector consisting of the probability of an observation, and a pose for that observation. During the training phase of the capsule networks, the weights are determined by an iterative algorithm called dynamic routing. In dynamic routing, we transform the vectors of an input capsules with a transformation matrix to form a vote, and group capsules with similar votes. Those votes eventually becomes the output vector of the parent capsule. A diagram of capsule architecture is shown in Figure 2.

Capsule networks make use of a high-dimensional transformation matrix, that requires a lot of memory. This limits the size of the images that can be used in the capsule network method. Moreover, the size of the weight matrix is fixed based on the input size, and the same network cannot be used for different size of images. LaLond and Bagci suggested a solution to those problems at 2018 [6], by expanding traditional capsule network into capsules that based on convolution operations.

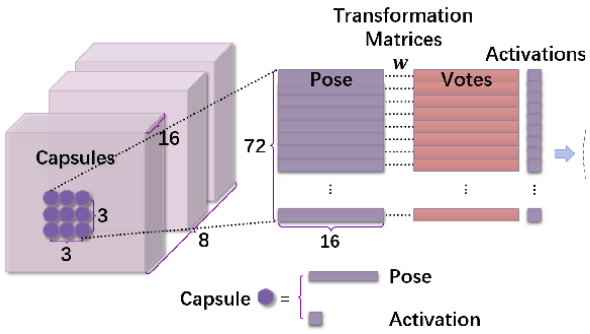

Fig. 2. A diagram of capsule architecture: input capsule units, features ("pose"), transformation matrix that forms a vote and output capsule units.

They extend the idea of capsules, and propose the concept of convolutional and deconvolutional capsules layers. Using convolution operations allowed the network to handle larger image sizes (512 x 512) as opposed to baseline capsules (typically less than 32 x 32). They also expended the dynamic routing algorithm with with locally-connected dynamic routing. We detail the local dynamic routing algorithm in Algorithm 1.

Our classifier network was built from convolutional capsules building blocks, due to the fact that capsules are better able to capture the spatial realtion between features than standard CNN (in our case: tumors, skull, and other brain parts). We belive that using convolutional capsule building block will allow us to reduce the number of parameters of the network, while preserving the performance of a deeper network.

B. Pix2Pix

Conditional Generative Adversarial network was first introduced in 2014 by Mirza and Osindero [10]. In the Conditional GAN (cGAN), the generator learns to generate a fake sample with a specific condition or characteristics (such as a label associated with an image or more detailed tag) rather than a generic sample from unknown noise distribution. Image-to-image are conditional GAN networks that are able to learn the mapping from input image to output image. The Pix2Pix Generative Adversarial network was first introduced in 2016 by Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros [8]. The paper presented an approach to train a deep convolutional neural network for image-to-image translation tasks. The careful configuration of architecture as a type of image-conditional GAN allows for both the generation of large images compared to prior GAN models (e.g. such as 256x256 pixels) and the capability of performing well on a variety of different image-to-image translation tasks.

Although our brain MRI dataset is considered large compared to other MRI's image datasets, it is still smaller than other datasets that are used to train neural networks. Our generative network based on the image-to-image network (Pix2Pix) to synthesize additional data, by transferring masks of tumors and skulls into MRI's images. We relied on existing implementation of the pix2pix and modified it to work with the brain MRI images in the brain MRI dataset. We changed the

input mask to 1D channel (instead of 3D channel) and modified the network so it could handle image size of (512x512) instead of (256x256). The current model of pix2pix doesn't take any noise as input, so that the output is deterministic and set for the same condition label. We expanded the model and fed the generator with a latent noise vector to control its output variations and to increase the diversity of the generated MRI images.

III. Setup

A. Dataset

For all our experiments, we used the 'Brain Tumor' dataset, proposed by Cheng Jun et al[3]. The dataset contains a 3064 T1 weighted and contrast-enhanced brain MRI's images , from 233 patients, and includes three classes of tumors: Glioma (1426 images), Meningioma (708 images), and Pituitary (930 images) tumors. The images were taken in different planes: sagittal (1025 images), axial (994 images), and coronal (1045 images) planes. Each image was given with a patient ID, tumor mask and a class label. The examples of different types of tumors, as well as different planes, are shown in Figure 3. The tumors are marked with a red outline.

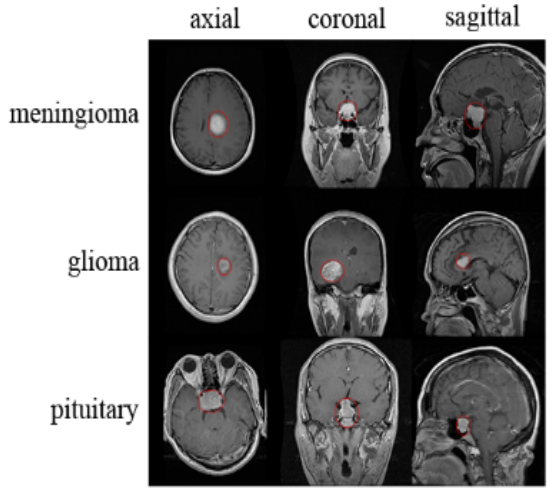

Fig. 3. Example of three classes of brain tumors: Glioma, Meningioma, and Pituitary, taken from different planes: axial, coronal, and sagittal.

B. Data Preprocessing

The conditional generative models are trained on labels and brain MRI images. The labels contains a mask of the tumors and skulls. The masks of the tumors are given within the 'Brain Tumors' dataset. We extracted the skulls mask from MRI images in the following steps:

- Apply adaptive binary threshold using Otsu's method.
- Low-pass filter to fill the innner-skull region
- Additional image closing to fill the remain holes and attain brain morphological integrity.
- Soft the edges using a low-pass filter
- Apply another binary threshold of 0.5

An example of a label, consists a tumor mask and extracted skull mask, is shown in Figure 4.

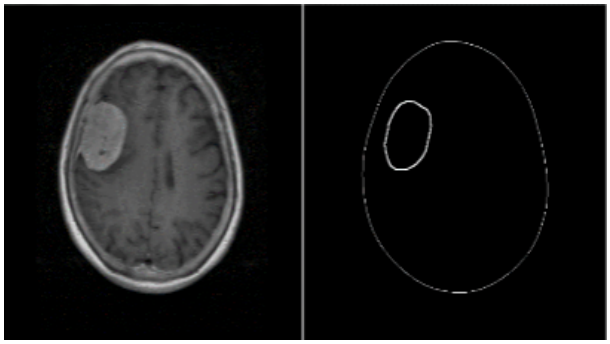

Fig. 4. An example of a label, consists a tumor mask and extracted skull mask, in the way described in the data preprocessing part.

C. Hardware

The generative networks were trained and tested on a single graphical processing unit (GPU), CUDA device, GeForce GTX 1650. The classification networks were trained and tested on a single graphical processing unit (GPU), CUDA device, GeForce GTX 1080 TI.

IV. Methodology

Our solution combines synthesise of additional data using a conditional generative model, with a new classifier architecture that is built upon convolutional capsules building blocks. The combination of capsule classifier and diverse data, will cause the network to better learn the spatial relations between the brain parts and the tumor, and to classify the tumors with less complex network architecture.

A. Generative Model

Generative Adversarial Networks (GANs) are generative model that map latent vector into images [11].

$$z \sim P_z\left(z\right)$$

Conditional generative model (cGANs) [10] map images from a latent vector z and a conditional setting. We used a conditional GAN for image-to-image translation, as the mask of the tumors and skulls were inputted as a condition for the generator, and the original MRI image were used as ground truth for the generated image.

$$G : \{c_x\} \rightarrow [x]$$

Our generator architecture is based on the U-net-style[9] encoder-decoder network and built from 8 upsampling, 8 downsampling and 1 bottleneck layers. The architecture is designed to output an image that is aligned with the structure of the input. Moreover, we use skip connections to encode both low and high level features. The pix2pix original generator implementation was of U-Net 256 architecture in which the bottleneck component is of size 1x1. We used the same implementation, thus in our case it will be reduced to bottleneck component of size 2x2. In order to generate a distribution of

synthetic images, we additionally augment our network with a latent vector

$$z \sim N_{100}(0, 1)$$

this vector is passed through a fully-connected layer of size $100x512^2$, with the output reshaped to be the same spatial dimensions as the input image $512x512$. This is then concatenated with the input mask label as an additional channel, so that the input into the first convolutional layer is of size [B,2,512,512], where B is the batch size. By fixing the input mask label and sampling different latent vectors, we wanted the network to produce varied images underline the same condition. The concatenation of the latent vector and the input mask label is shown in Figure 5.

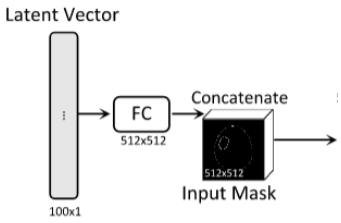

Fig. 5. Concatenation of the latent vector and the input mask label.

The full architecture that we used of the generator, is shown in Figure 6.

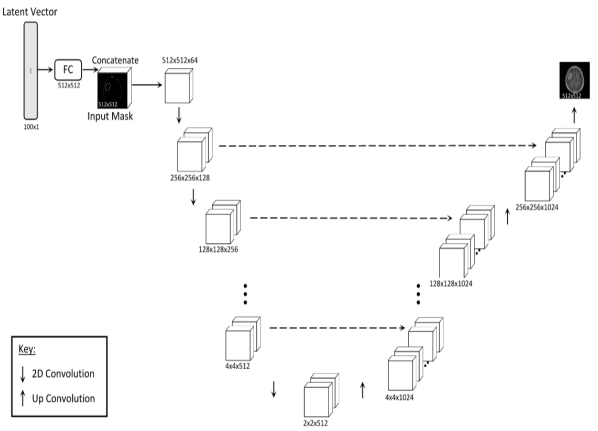

Fig. 6. Architecture of the generator of the generative model.

The conditional GAN also has a discriminator model. Th generator is trained to generate samples that fool the discriminator. The discriminator is shown both real and synthetic image-label pairs and is trained to distinguish between then via a binary classifier.

$$D : \{[x, c_x], [y, c_y]\} \rightarrow \{0, 1\}$$

For the discriminator, we used the PatchGAN classifier that was used in the pix2pix paper, with some modifications. Enlarging the GANs causes the patch output of the discriminator to be 4 times larger (62x62 output instead of 31x31). A $(62x62)$ PatchGAN will classify $(62x62)$ patches of the input image as real or fake. A convolution is applied after the last layer to map to a 1-dimensional output, followed by a Sigmoid function. BatchNorm is not applied to the first layer. All ReLUs are leaky, with slope 0.2.

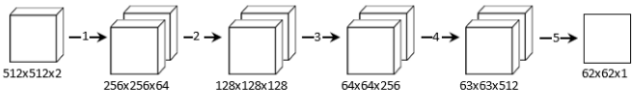

Fig. 7. Diagram of the discriminator model.

Our generative model block diagram is shown in Figure 8. Training is is made to minimize the min-max loss, where the objective is to find an equilibrium for both models:

$$\min_G \max_D L_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x \sim P_{data}(x), z \sim}, [\log(1 - D(x, G(x, z)))]$$

In order to encourage the final output to better adhere to the structure of the label, we also minimise an L1 loss between synthesised and real images with the corresponding label:

$$\min_G L_1(G) = E_{x,y \sim P_{data}(x,y), z \sim P_z(z)}[||y - G(x, z)||_{L1}]$$

The final objective loss is the value function V (G,D):

$$\min_G \max_D V(G, D) = L_{cGAN}(G, D) + \lambda L_1(G)$$

where $\lambda = 100$. We tested different λ values to check whether decreasing λ could generate more diverse images, for different latent vectors, while still keep the appearance of brain MRI image realistic.

B. Classification Models

Convolutional capsule layers take input a, of size $[B, I, W_i, H_i, C_i]$ and output $\widehat{u}$ of size $[B, I, W_j, J, C_j]$, where B=batch size, I=number of input capsules, CI=number of input channels, W=width, H=height, J=number of output capsules and CJ=number of output channels. In the convolution step, the kernels $k_w X k_h$ are shared between the input capsules in order to reduce the number of the parameters. For each layer of the network, the activations of the child capsules are routed to all parent capsules.

LaLonde and Bagci (2018) also introduce the local dynamic routing algorithm to accompany convolutional capsules [6]. In the method of convolutional capsules, each spatial location in the input capsule is routed to the same spatial location in the output capsules, using a spatial kernel. The spatial kernel is formed from parameter b and normalised to form vectors c_{ij} using the softmax function

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_i \exp(b_{ij})}$$

where b_{ij} is updated in every routing iteration. The local dynamic routing includes taking the dot product of the output convolution $\widehat{u}$ with c_{ij} over the input capsules. Each spatial

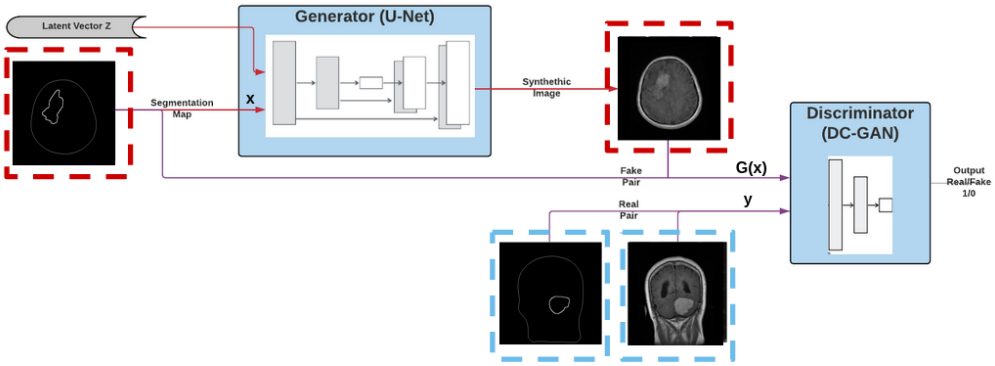

Fig. 8. Example of three classes of brain tumors: Glioma, Meningioma, and Pituitary, taken from different planes: axial, coronal, and sagittal.

location in the input capsules is routed to the corresponding location in the output capsule. The output of the dot product,s_j, is passed through a nonlinear squash function. The purpose of the squash function v_j is to normalise the output between 0 to 1 and to ensure that the vector direction is retained. For the update, b_{ij} is incremented with the dot product of v_j with $\widehat{u}_{ij}$. The local dynamic routing is fully described in Algorithm 1.

Algorithm 1: Convolutional Capsules + Local Dynamic Routing

Input: a, capsules in layer l; l, layer; r, iterations; bias; weight

Output: v_j, capsules in layer $(l+1)$

$\hat{u}_{I\times J\times C_J} \leftarrow bias_{J\times C_J} + \sum_{n=0}^{C_l} weight_{J\times C_J,n} * a_n$

for all capsules i in layer l and capsules j in layer $(l+1)$: $b_{ij} \leftarrow 0$

for l **to** r **do**

  for all capsules i in layer l and capsule j in layer $(l+1)$: $c_{ij} \leftarrow softmax(b_{ij})$

  for all capsules j in layer $(l+1)$: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{ij}$

  for all capsules j in layer $(l+1)$: $v_j \leftarrow squash(s_j)$

  for all capsules i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{u}_{ij} \cdot v_j$

end

Fig. 9. Local dynamic routing algorithm.

The different classes of the tumors differ in their shape, affected area, and their spatial relation to other brain parts, such as skull and brain tissues. The classifier model was built quite similar to the discriminator of the generative model. However, there are a few key differences:

- Since our network is a classifier, we have a single image which is an input to the network.
- The image passes a convolutional layer into a capsule unit. Each capsule meant to represent different features, such as tumor, skull, brain tissue, etc.
- In addition, each layer of the network is split into blocks of capsules, and each capsule has a several number of channels.
- Finally, the output capsule passes another convolutional layer into a set fully connected layers for the classification.
- We used softmax to map the non-normalized output of a network to a probability distribution over predicted output classes.

The architecture of the capsule classifier is shown in Figure 10, and had a total number of 3,827,660 parameters. We also examine the state-of-the-art Resnet50 network which had a total number of 31,906,883 parameters. For both network we used 8x data augmentation and 7-fold cross validation.

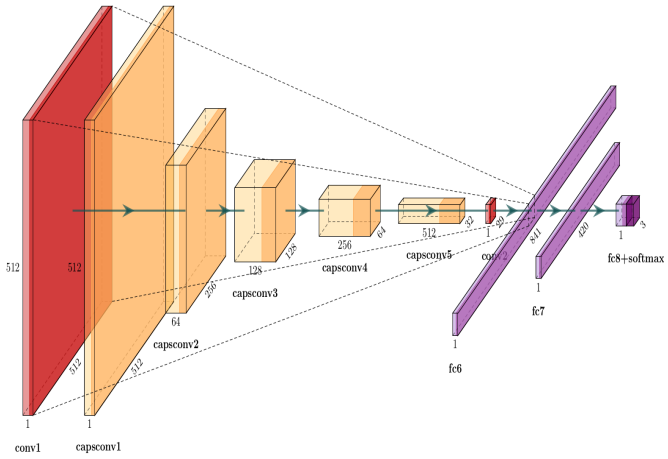

Fig. 10. The architecture of the capsule classifier.

V. Experiments and Results

Our goal was to test whether using the synthetic images could improve the classification task upon training with real images. We evaluated the performance in both qualitatively and quantitatively ways.

Qualitative Analysis:

- Synthesise of different images for the same skull label.
- Linear interpolation of the latent space.
- Compare the last layer features, produced from the same label.

Quantitative Analysis:

- Train separate networks, on different datasets: real and synthetic images.
- Performance was measured using: accuracy and F1 scores.

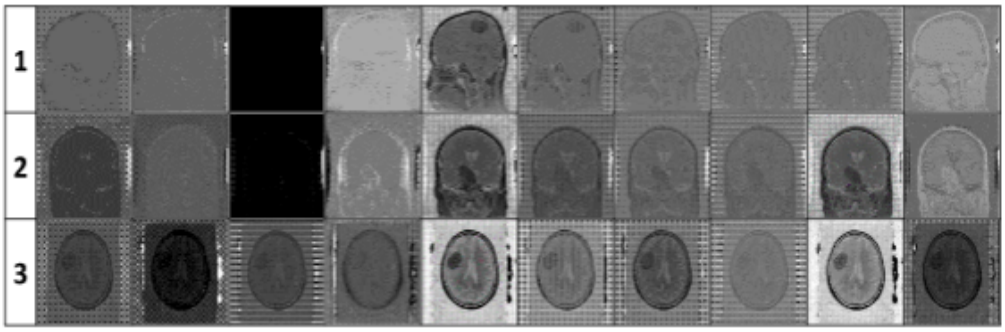

Fig. 11. The features (activations) of the last capsule layer for the Meningioma generative network.

A. Evaluation of Generative Model

We trained three conditional generative models, one for each class of tumor: Glioma, Meningioma, and Pituitary. The Meningioma and Pituitary tumors were more noticeable and thus the generative networks were able to generate more realistic images than the Glioma tumor generative network, as demonstraded in Figure 12. For each network, we generated

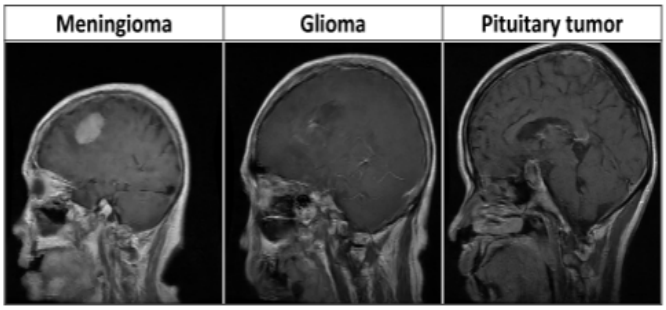

Fig. 12. Example of three classes of brain tumors: Glioma, Meningioma, and Pituitary, that generated with our generative network.

new brain MRI images underlying the same skull geometry, while reshaping the tumors mask. The tumors shape modification was achieved by applying rigid body affine transformations: scaling, rotation and shear. We synthesised different images for the same skull label from each network, as demonstrated in Figure 13, and found out that generative models learned successfully the tumor and the skull classes, as the images still appear very realistic. Overall, we found this method very effective at increasing the amount of trainable data, as the new images includes new variations of tumor, skull and other brain parts.

We also display the features (activations) of the last capsule layer, demonstrating the variety of features learned by the network. In addition, we found that individual capsules appear to group similar features. The features of the last capsule layer, of the Meningioma generative network, are shown in Figure 11. Encoding the tumors, skull and brain parts classes could improve the classification of unbalanced data, as we could match the number of images in each class with generation of additional images.

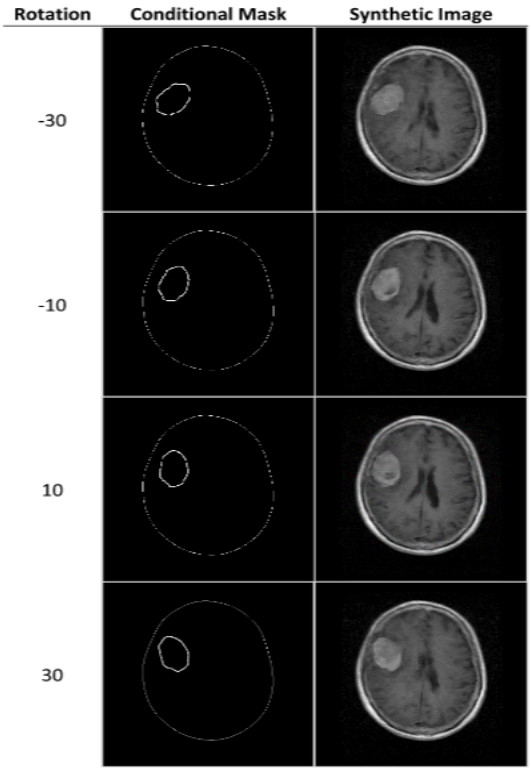

Fig. 13. Generation of different images underline the same skull label from each network, for Meningioma tumor. The generated images appear realistic.

Finally, we synthesised different images underline the same label from each network, each image was generated for different latent vector. We intended to introduce randomness through latent vector and to increase the variation of the synthesised images. We examine the effect of the latent vector for cGAN L1 coefficient $\lambda = 0.1, 100$. We found that generative models was able to make minor alterations to the images, as demonstrated in Figure 14. We conclude that our model wasn't able to learn the noise class through a latent vector, however, we got diverse images from changing the tumor class underline the same skull geometry and it was enough to increase the number of labelled images in the dataset.

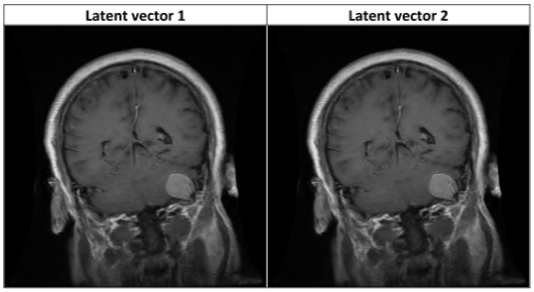

Fig. 14. Generated Meningioma brain MRI images for the same condition, and different latent vectors

B. Evaluation of Classification Models

The same capsule classifier architecture was trained on different datasets, and evaluated on the test set of 450 brain MRI images (15% images for each class). We also compared the result to the state-of-the-art deep complex Resnet50 classifier. We tested the following:

- Capsule classifier trained on real data.
- Capsule classifier trained on synthetic data.
- Capsule classifier pre-trained on synthetic data and than trained on real data.
- Resnet50 classifier trained on real data.

The real images dataset consist of 2,614 images and additional 7,000 images of size 520x520 were synthesised for the synthesised images dataset. The classification performance were measured using accuracy and F1-score, which is the harmonic mean of the precision and recall. The results are shown in Table 1.

Classifier	Images	Pretrained	Accuracy	F1-score
Capsule	Real Data	No	0.85	0.86
Capsule	Generated	No	0.87	0.86
Capsule	Real Data	Generated	0.93	0.92
Resnet50	Real Data	No	0.94	0.94

Table1 Result Table

Overall, we conclude that training a capsule classifier from scratch on generative synthetic data leads to comparable results to training on real data, since they lead to similar results. Pre-training capsule classifier on synthetic dataset and than train on real data was able to improved the performance significantly and was able to obtain similar result to the Resnet50 classifier model with 9x less parameters.

VI. Conclusion

In this paper we introduced a novel method which combines synthesise of additional data using a conditional generative model, with a new convolutional capsule architecture for classification. The best result of 93% accuracy and F1 score of 92%, was obtained when the classifier pre-trained on 7,000 synthesised images and then trained on the original data using a 7-fold cross-validation. Our method lead to comparable results to the state-of-the-art Resnet50 classifier, with 9x fewer parameters. We also showed that training on synthetic data performed relatively similarly to training on real data, and that pre-training on synthetic data and than training on real data improved the performance of the classifier.

Qualitatively, even though that the generative model haven't learned the noise class out of the latent vector, we managed to generate diverse images while reshaping the tumor mask for the same underlying skull geometry. A remarkable thing to note about the synthetic images is that as the tumor get closer to the center of the brain it, the gap between the lobes shrinks. We conclude that network managed to learn the anatomic relations between the tumor, skull and brain successfully. Moreover, the generative model have learned relevant features for our dataset, and appears to capture the distribution of both the tumor and brain. Due to the end, the generative model improved the classification accuracy of the capsule classifier that learns the spatial relations between the features of the image.

In future, we suggest to test the following topics:

- Using capsule conditional generative model, instead of convolutional cGAN. Image synthesis could be much more variable using Capsule cGAN, because it's features could capture both tumor, skull and noise classes.
- Increase the diversity of the brain MRI synthesised images by also reshape the skull mask.
- Examine different ways to insert noise into the network, including broadcasting noise, adding it to the bottleneck, or using dropout in inference as in pix2pix,

VII. Acknowledgments

Special thank to Cher Bass for publishing 'Image Synthesis with a Convolutional Capsule Generative Adversarial Network' [7] and inspired our work. We learned a lot from the paper about capsule networks, convolutional capsules and conditional GAN.

We also thank M. Bada for publishing the paper 'Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network'[3] that taught us about the biological background and helped us to define the problem better.

References

[1] American Society of Clinical Oncology (ASCO): Brain Tumor. Available online: https://www.cancer.net/cancer-types/brain-tumor/statistics. (accessed on 1 July 2020).

[2] Adrian P. Brady. Error and discrepancy in radiology: inevitable or avoidable?. Insights Imaging. 2017 Feb; 8(1): 171182.

[3] Milica M. Bada et al. Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network. 2020.

[4] Cheng, J. Brain Tumor Dataset. 2017. Available online: https://doi.org/10.6084m9.figshare.1512427.v5 (accessed on 1 July 2020).

[5] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In Advances in Neural Information Processing Systems, pages 38563866, 2017.

[6] Rodney LaLonde and Ulas Bagci.Capsules for object segmentation.arXiv preprintarXiv:1804.04241, 2018.

[7] C. Bass, T. Dai, B. Billot, K. Arulkumaran, A. Creswell, C. Clopath, V. De Paola, and A. A.Bharath, Image synthesis with a convolutional capsule generative adversarial network, Medical Imaging with Deep Learning, 2019.
[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation withconditional adversarial networks.arXiv preprint, 2017.
[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedi-cal image segmentation. InInternational Conference on Medical image computing and computer-assisted intervention, pages 234241. Springer, 2015.
[10] Mehdi Mirza, Simon Osindero. Conditional Generative Adversarial Nets.arXiv preprint arXiv:1411.1784, 2014.
[11] Ian Goodfellow.Nips 2016 tutorial: Generative adversarial networks.arXiv preprint arXiv:1701.00160, 2016.

VIII. Appendix A: Code

The code for our method can be found at the project github page. The code that we written [@pytorch] includes:

- Data preprocessing [@Matlab].
- Modified cGAN with a latent vector.
- Capsule classifier model.
- t-SNE.
- Dataloader.
- Cross-validation.
- Train and evaluate different classifiers.
- Resnet50 1D classifier, with an option to freeze layers.

We used the following code parts:

- We based our cGAN on Pix2Pix pytorch implementation.
- Data augmentation (rotate) for the classifiers.
- Convolutional capsule building block and dynamic routing pytorch implementation from CapsPix2Pix paper.

IX. Appendix A: Classification Details

We use SGD as our optimiser, with learning rate=0.0003 ,momentum=0.9; We decay the learning rate to 0.0001 by the end of training. A detailed information about the network architecture can be found at the project github page. We show the confusion matrices for the best results experiments, and visualisation of the t-SNE of the features from the last layer, before the classification operation.

Capsule classifier, CM

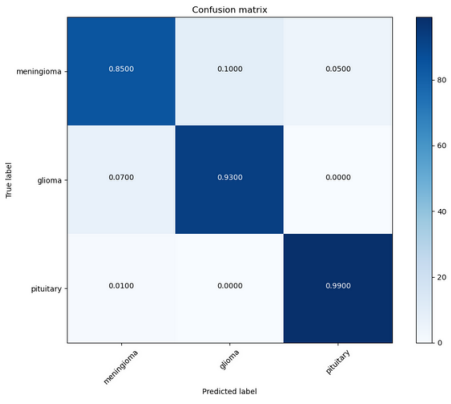

Fig. 15. The confusion matrix of the classifier who pre-trained on synthetic data and than trained on real data - accuracy of 93%

Capsule classifier, t-SNE

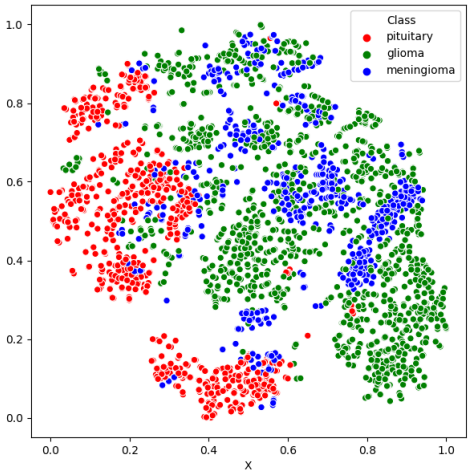

Fig. 16. The t-SNE of the classifier who pre-trained on synthetic data and than trained on real data - accuracy of 93%

Resnet50 classifier, CM

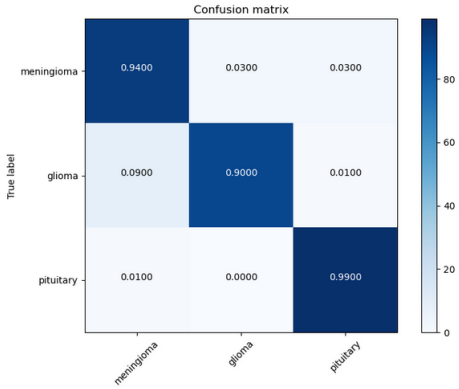

Fig. 17. The confusion matrix of the Resnet50 classifier- accuracy of 94%

Resnet50 classifier, t-SNE

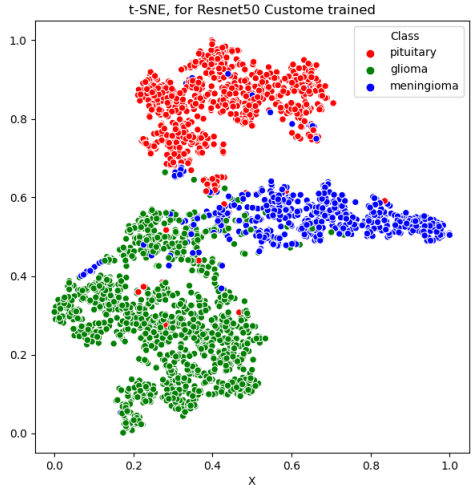

Fig. 18. The t-SNE of the Resnet50 classifier- accuracy of 94%