Deep Learning for Stroke Detection

Giuseppina Carannante
ECE Department,
Rowan University
Glassboro, NJ
carannang1@rowan.edu

***Abstract*— This project was aimed to correctly detect stroke from brain CT scans using Convolution Neural Networks (CNNs). Head computed tomography (CT) scan is the current standard to diagnose head trauma and stroke symptoms. When planning medical and surgical treatments, it is very important to aim for an early and timely detection to minimize long-term consequences. This need has driven many studies to implement and improve Deep Learning techniques for an automated classification of stroke symptoms. The main objective will be to aid radiologists providing supplementary information for an early diagnosis. For this study, the data was given in form of DICOM files, and labels were provided. Hence, a supervised approach was used to solve the problem. For medical image classification, CNNs have been proved to be very successful. Thus, for this project, a CNN based on an existing architecture, SmallerVGGnet [1], has been implemented for the given classification task. Different training sessions were performed altering slightly the architecture of the model. Hyperparameters were tuned to achieve a better accuracy and a lower loss. Overall, the model achieved a 90% accuracy on the training and validation sets while the training and validation losses were roughly constant around a 20% value showing no sign of overfitting. However, the model still needs improvements since some classes are still not correctly identified and others have low scores. First, we need to observe that only part of the given 500 patients was used to train the model reducing the size of the dataset. Therefore, to achieve better results, more data need to be collected for training. In fact, some classes were under-represented, and the model did not show to perform well on these while others were not used during training, thus the model will never be able to predict those.**

Keywords*— *Convolution Neural Network, Classification, CT brain images

I. Introduction

Stroke is among the leading causes of impairment and death all over the world. An accurate and expeditious detection is critical for the management of acute stroke since delays in patient care reduce the chances of positive outcomes. For the identification of stroke symptoms, the presence of an expert radiologist is needed in every hospital facility, but this is not always possible. Moreover, the current process to correctly classify head trauma requires the radiologists to analyze large volumes of images making the diagnose subject to mistakes and the image interpretation may be subjective leading sometimes to contrasting opinions. However, the effectiveness of stroke treatments decreases as time passes making the identification of high-priority patients very important. For all these reasons, an automated tool for rapid classification of head trauma represents a compelling solution to assist radiologists and facilitate prompt treatment responses when needed. In fact, an automated process could help in situations of unavailability of neurological expertise since the radiologist presence will not be required. On the other hand, these tools could also accelerate and improve the existing classification-treatment process since the objective response could be used to find consensus in case of radiologists having differing opinions and to quickly identify among all cases those that require urgent attention.

Deep Learning Convolution Neural Networks (CNNs) have the capacity to capture and extract features from raw data and they appear to be very useful in many problems involving image content. In particular, some existing algorithms have been proved to be very successful in classification and segmentation tasks in the medical field, achieving state-of-art results. A CNN is built by stacking together many layers of convolutions where each neuron captures an overlapping region of the visual field similarly to what happens in the biological vision process. In particular, they are very useful since they do not rely on feature-engineering, but rather features maps are learnt with a forward-backward propagation process.

As mentioned, the current process to diagnose stroke requires the radiologist to go through many images. A CT scan is made of multiple DCM files, each containing one image that corresponds to a single brain slice. All slices stacked together enable radiologists to have a '3-D image' of a patient brain. CNNs can easily be implemented to take CT scans as inputs. The data used for training is of the type $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ where $\boldsymbol{x}^{(i)}$ will represent our three-dimensional input and the vector $\boldsymbol{y}^{(i)}$ the target label that our model will try to predict. Therefore, this study is meant to investigate a supervised automatic process employing Convolutional Neural Networks (CNNs) to correctly identify head trauma and classify CT data under the correct label.

II. Methods

A. Preprocessing Phase

For this project, a CNN able to work with multiple classes and multiple labels is needed. The given dataset is composed of 500 patients CT scans and a labels file with the responses of three radiologists. The images used to train the proposed architecture were chosen for patients whose labels where not showing differing opinions between radiologists. Therefore, not all provided CT data were used during training. CT scans were given as DICOM files, each containing 2D arrays with pixel intensities in Hounsfield Units (HU)[2]. HU aren't in the standard 0–255 interval; they range from -1000 HU for air, to 0 HU for distilled water, and >10,000 HU for metals, thus, an initial preprocessing stage was needed. Then, data were randomly partitioned into training and validation sets using 80% of the data for training and the remaining 20% for validation. Augmentation was used to generate additional data to prevent overfitting due the small size of the training dataset.

B. Network Architecture

For this study, there was a total of 14 classes provided: ICH, IPH, IVH, SDH, EDH, SAH, BLEEDL, BLEEDR, CBLEED, FRAC, CFRAC, OFRAC, MASSEFFECT, and

https://github.com/GiuseppinaC/ECE_Project

MIDSHIFT. One CT image could be associated with multiple labels; for example, a patient could present signs of a fracture and be suffering of hemorrhage. For this reason, the target label **y** has multiple positive entries: 0s for incorrect classes and 1s for correct classes. Hence, a multi-label classification architecture was implemented using sigmoid activations in the last layer:

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

Each neuron in the output layer will give a value between 0 and 1, predicting if the input belongs to a class or not. The Smaller VGGnet architecture, a more compact version of VGG net, was implemented for this classification task. Being mainly based on VGGnet, the used architecture is characterized by the presence of only 3x3 convolution kernels, and an increasing number of filters as the depth of the model increases. Moreover, for all the convolution layers Padding is set to SAME to ensure that the size of the output feature map is the same as the size the input feature map. This Deep Learning Model is constructed as follow: the first block is made of a convolution layer with 32 kernels, followed by rectified linear unit (ReLU) activation function and Batch Normalization; then, Max-Pooling is applied. In the successive block, the following sequence of operations is repeated twice: convolution with 64 kernels, ReLU and batch normalization; Max pooling is applied only once at the end of this sequence to account for dimensionality reduction. This process of Stacking multiple sequences of convolution and ReLU operations followed by batch normalization is applied again prior to reduce spatial dimensionality with one Pooling operation. The only difference with the previous block is in the number of kernels increased to 128. Moreover, max-pooling operation is changed from a size of 3x3 to a 2x2 to ensure not to decrease the dimensionality too much. In this model, dropout is applied after every pooling operation with a probability p equal to 0.25 to reduce the risk of overfitting. Additionally, there is one Fully Connected (FC) layer with 1024 units, ReLU activation, and dropout with probability 0.5, followed by the output layer with the number of units equal to the number of classes to predict. As explained above, sigmoid activation function is used in the output layer to give the scores for the predicted classes allowing the network to perform multi-label classification.

As it has been well established, a CNN tries to approximated a function f^{w} such that

$$\boldsymbol{y}^{(i)} \approx f^{\mathrm{w}}(\boldsymbol{x}^{(i)})$$

This function f^{w} depends on the parameters of the model, the kernel weights w_{ij}. They are learnt using a supervised optimization approach. w_{ij} are randomly initialized and backward propagation ensures that they are updated so that the overall loss function is minimized.

```
Total params: 8,778,604
Trainable params: 8,775,532
Non-trainable params: 3,072
```

Fig. 1. Total Model parameters for the proposed architecture.

The problem can be written as

$$\underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(f^{\mathrm{w}}(\boldsymbol{x}^{(i)}), \boldsymbol{y}^{(i)})$$

where the loss $\mathcal{L}$ is a suitable function that when minimized ensures the output of the network, $\hat{y} = f^{\mathrm{w}}(\boldsymbol{x})$, is as close as possible to the "ground-truth" value $\boldsymbol{y}$. Given the nature of our problem, the loss chosen for the optimization process was the binary-cross entropy function:

$$\mathcal{L} = -\sum y \log \hat{y} + (1 - y) \log(1 - \hat{y})$$

An initial training session consisting of 100 epochs was done using Stochastic Gradient Descent (SGD) with momentum and an initial learning rate of 0.1. Then, for the last training, optimization was implemented using the Adam method with a learning rate of 0.001 leading to very similar results.

III. Results

The model performs better in classifying some classes more than others. After different training sessions the following results were found:

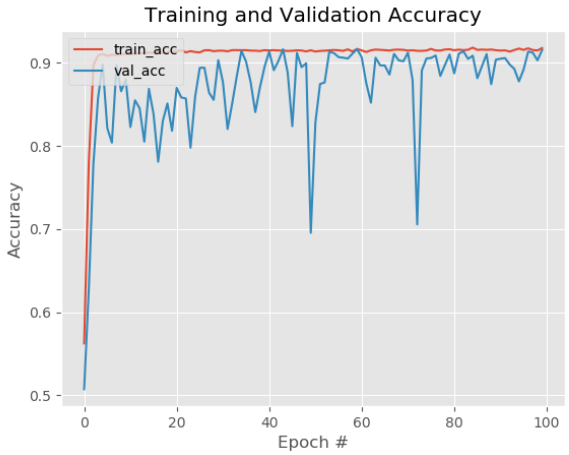

Fig. 2. Accuracy plot for training and validation data.

Fig. 3. Loss plot for training and validation data.

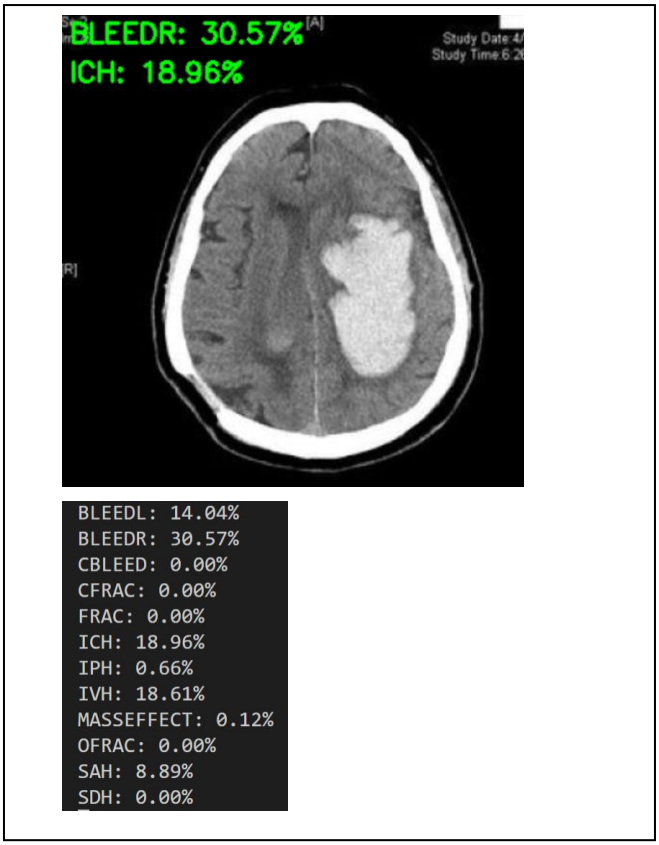

Fig. 4. ICH classification example. The model predicts this class correctly, but the percentage is not very high.

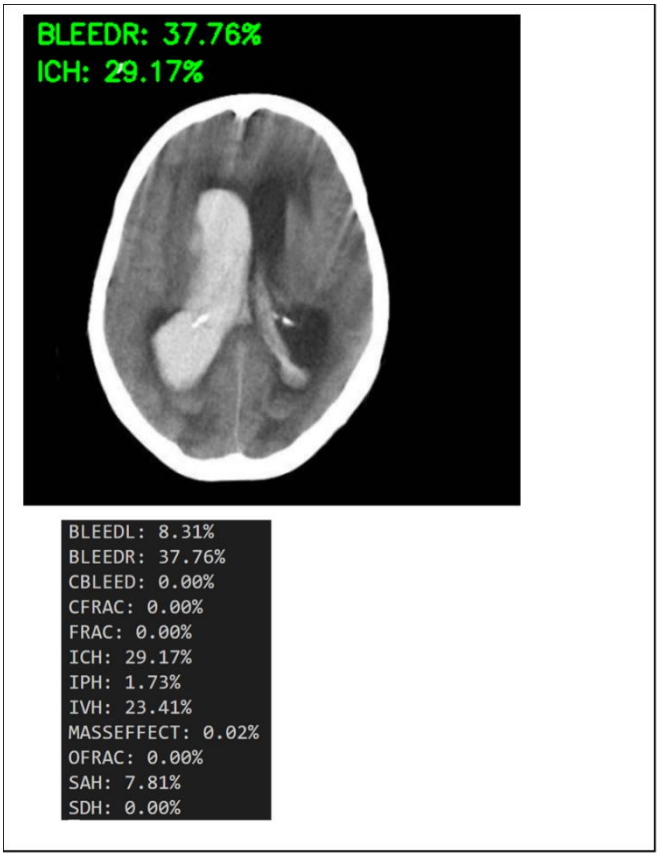

Fig. 5. IVH example. The model is identifying ICH with higher percentage while IVH has only 23.41% reasong being the presence of a white mass for both ICH and IVH.

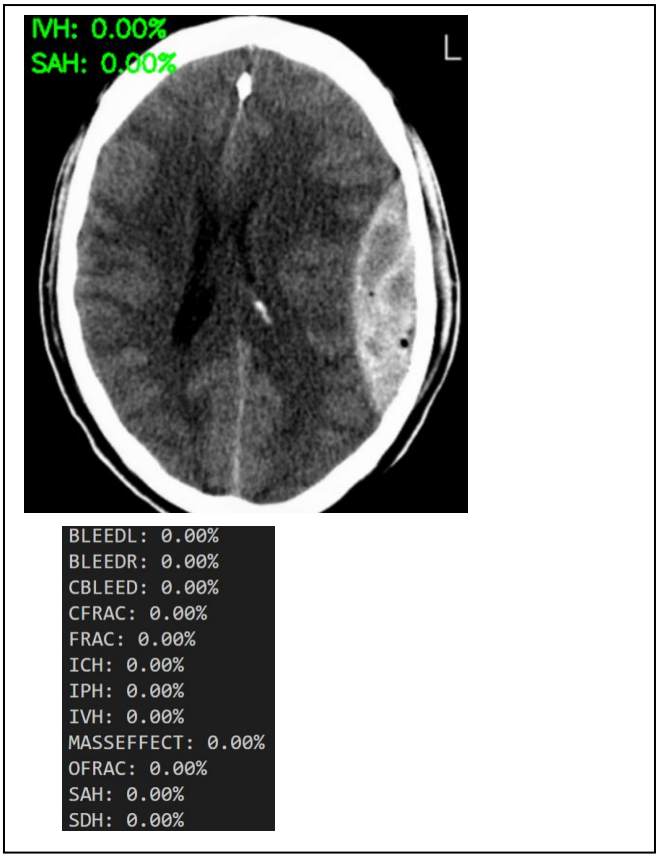

Fig. 6. EDH example. This class was not used during training and the model is clearly not able to categorize the image.

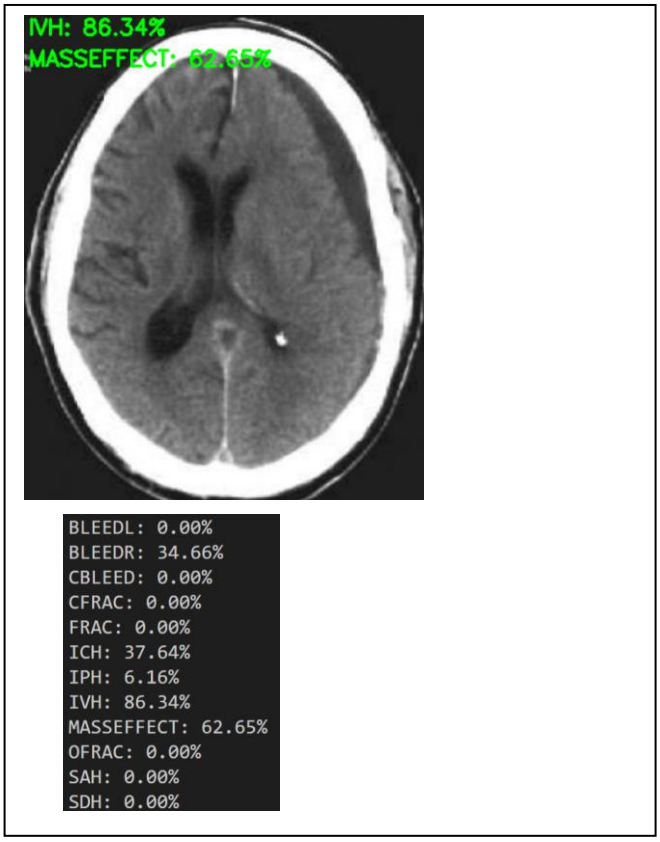

Fig. 7. MASS EFFECT. The model is correclty identifying a Mass Effect, but at the same time there is a missclassification since there is no evidence of IVH while it should be also labelling this image as SDH.

IV. Discussion

Slightly different architectures based on VGGnet were investigated for this dataset. The first trial was conducted leaving invariant the provided architecture of SmallerVGGnet. Only the last layer activation function was changed from softmax to sigmoid function adapting the model to a multi-label classification. The first training session consisted of 75 epochs, and there was no sign of overfitting, but a better model was needed since the results were not satisfying enough: the model was not predicting correctly all classes. So, a more complex architecture was implemented to get better results. In a second training session, an additional sequence of Convolution-ReLU was stacked after the first layer with the same number of filters, and an additional fully connected layer at the end. Dropout was incremented to 0.5 for all layers to reduce the possibility of overfitting. However, these modifications did not provide any additional improvement to the model. Therefore, to reduce the parameter space size the initial conditions were re-established. Hyperparametes were tuned more than once. As mention, also vary optimization techniques were investigated to reduce the loss. The provided results were obtained using ADAM optimization method. All the training sessions produced similar results: the training and validation accuracies were reaching values around 90% while the training and validation losses were constant around 20%. One training was done using class weights. The results of the previous training sessions showed that some classes were predicted better than others. Even altering different parameters and structure of the model, some classes were still incorrectly identified. Therefore, to handle the presence of unbalanced data, class weights were used to maximize accuracies on the under-represented classes. The weights were chosen in a simple way by setting them equal to the inverse of the class size. By doing that, loss was penalized more for minority-classes. However, the algorithm's accuracy did not significantly differ from previous training sessions, but there was a slight sign of over-fitting. In conclusion, all these trials showed that to improve the model ability to correctly classify all classes a larger representative dataset is required. Furthermore, the classes used for training are only 12 while original classes were 14. The model was trained using the labels provided by three radiologists and the labels EDH and MIDSHIFT had no CT scans with consensus among radiologists. Therefore, for the model to be able to predict correctly all classes, it must be trained on these labels too; hence, more CT scans are needed.

V. Conclusion

This study was concerned with the classification of CT images for stoke detection. The current process for stroke symptoms diagnoses requires radiologists to go through a large volume of images. To deliver a faster and better care when neurological expertise is unavailable or when a subjective interpretation can create contrasting diagnoses, an automated process would be preferable. CNNs demonstrated to be successful in image classification extracting feature from raw images without requiring a user to directly visualize the images. In this project, different methods were investigated, SmallerVGGnet was implemented and modified for this multi-label classification task. A relatively high accuracy was reached: around 90% for training and validation, but still some classes were misclassified. Thus, as suggested above, this model needs improvements and more data are required to achieve better classification results on all the 14 classes. Therefore, different architectures should be investigated, and different model structures used in the future.

References

[1] https://www.pyimagesearch.com/

[2] https://medium.com/stanford-ai-for-healthcare/superman-isnt-the-only-one-with-x-ray-vision-deep-learning-for-ct-scans-290aaa7ba5c1.

[3] Xiaohong W. Gao, Rui Hui , Zengmin Tian, "Classification of CT brain images based on deep learning networks," October 2016.

[4] Sasank Chilamkurthy, Rohit Ghosh, Swetha Tanamala, Mustafa Biviji,Norbert G Campeau, Vasantha K. Venugopal,Vidur Mahajan, Pooja Rao, Prashant Warier, "Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study" October 2018.

[5] Michael A. Bruno, Eric A. Walker, Hani H. Abujudeh, " Understanding and Confronting Our Mistakes: The Epidemiology of Error in Radiology and Strategies for Error Reduction" 2015.