UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Spontaneous intracerebral hemorrhages detection using deep learning on CT scan images

Luis Adrián Erazo Erazo

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Ciencias de la Computación

Quito, 28 de diciembre de 2021

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Spontaneous intracerebral hemorrhages detection using deep learning on CT scan images

Luis Adrián Erazo Erazo

Nombre del profesor, Título académico **Noel Pérez, Ph. D**

Quito, 28 de diciembre de 2021

© DERECHOS DE AUTOR

Nombres y apellidos: Luis Adrián Erazo Erazo

Código: 00201091

Cédula de identidad: 1724251200

Lugar y fecha: Quito, 28 de diciembre de 2021

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

Las lesiones cerebrales son uno de los problemas médicos más graves que pueden sufrir las personas. Estas lesiones tienen una amplia gama; sin embargo, las hemorragias intracraneales son algunas de las más críticas. Si no se tratan a tiempo, pueden provocar daños cerebrales irreparables, discapacidad, cambios en el estilo de vida del paciente y la muerte. Sin embargo, dado el tiempo que tarda un radiólogo en analizar imágenes cerebrales, no siempre se ofrece un diagnóstico y tratamiento en el momento adecuado. En este contexto, este estudio se centra en ayudar a los médicos a diagnosticar una hemorragia intracraneal a tiempo mediante el uso de redes neuronales convolucionales para procesar estas imágenes cerebrales. Dado que la tomografía computarizada es una de las tecnologías más asequibles debido a su precio y disponibilidad, se utilizó este tipo de exploración cerebral. Inspirándose en el modelo U-Net, se entrenaron y evaluaron dos modelos de arquitectura profunda. Estos modelos tenían como objetivo segmentar la región de la imagen en la que estaba presente una lesión y resaltar como una máscara superpuesta sobre la imagen cerebral. Se utilizaron tomografías computarizadas de 82 pacientes para entrenamiento, validación y pruebas. Las imágenes se escalaron a 256x256 píxeles antes de usarse como entrada para los modelos. Cada una de las arquitecturas se entrenó con diferentes tamaños de lote y el número de épocas. El mejor modelo obtuvo un valor de 0,85 en la intersección sobre la métrica de unión, 0,89 en el coeficiente de dados y 99,91% en precisión.

Palabras clave: Aprendizaje profundo, U-Net, Red neuronal convolucional, Segmentación de imágenes, Hemorragia intracerebral, Tomografía computarizada.

ABSTRACT

Brain injuries are one of the most severe medical problems that people can suffer. These injuries have a wide range; however, intracranial hemorrhages are some of the most critical. If not treated in time, they can lead to irreparable brain damage, disability, patient lifestyle changes, and death. However, given the time it takes for a radiologist to analyze brain images, a diagnosis and treatment are not always offered at the right time. In this context, this study focuses on helping doctors diagnose an intracranial hemorrhage earlier through the use of convolutional neural networks to process these brain images. Since computed tomography is one of the most affordable technologies due to its price and availability, this type of brain scan was used. Inspired by the U-Net model, two deep architecture models were trained and evaluated. These models aimed to segment the region of the image in which a lesion was present and highlight as a mask superimposed over the brain image. CT scans of 82 patients were used for training, validation, and testing. The images were scaled to 256x256 pixels before being used as input for the models. Each of the architectures was trained with different batch sizes and the number of epochs. The best model obtained a value of 0.85 in the intersection over the union metric, 0.89 in Dice coefficient, and 99.91% in accuracy.

Key words: Deep learning, U-Net, Convolutional neural network, Image segmentation, Intracerebral hemorrhage, Computerized tomography.

TABLE OF CONTENTS

INDEX OF TABLES

INDEX OF FIGURES

INTRODUCTION

According to the United States Centers for Disease Control and Prevention, in 2018, one in every six deaths from cardiovascular disease was due to stroke. After heart disease, brain stroke is the leading cause of mortality worldwide (Centers for Disease Control and Prevention, 2021). Furthermore, even if the person survives, most survivors are forced to live with a constant or long-term injury that affects their living standards considerably. This is because strokes cause brain tissue to die, leading to brain damage, disability, and death. To avoid these consequences, people with a stroke should receive treatment as fast as possible. However, due to lengthy doctor diagnoses, time-consuming tests and scans, and not-so-identifiable symptoms, people do not receive treatment on time. For this reason, any tool that allows brain strokes and hemorrhages to be diagnosed easier or faster is considered to be highly relevant nowadays.

Despite several existing imaging technologies, Computerized Tomography is the diagnosing tool that doctors most widely used for patients with possible strokes or hemorrhages. Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) are the two most prevalent brain imaging technologies that allow doctors to identify and diagnose strokes and hemorrhages in a patient. MRI is widely used; however, due to the higher cost and unavailability in some hospitals and clinics, this technology is surpassed by CT since this one is non-invasive and the least expensive. Other technologies are SPECT or XENON tomographies, Positron Emission Tomography (PET), Magnetic Resonance Spectroscopy (MRS), Functional Magnetic Resonance Imaging (f-MRI), and Carotid Ultrasound (CU). However, these imaging modalities necessitate a high operational cost and a well-trained operator, so they may not be available in many clinics and hospitals (Harpaz, D., Eltzov, E., Seet, R. et al, 2017). Additionally, almost every hospital has the service of CT scans and is one

of the fastest methods of diagnosing brain strokes and hemorrhages. This makes Computerized Tomography scans one of the most used brain imaging technology for this procedure.

Doctors need additional technological tools to help them make a faster and more reliable diagnosis when checking brain images of potential strokes or hemorrhages. Identifying stroke from CT scan pictures is the first step toward a patient's accurate diagnosis. Initially, the images are sent to a radiology specialist to determine the sort of stroke. Following that, patients are examined physically and manually, and therapy is initiated depending on the results of the manual examination. When a significant number of patients with stroke symptoms arrive at the hospital on the same day, it might be difficult to give proper therapy on time. As a result, because this manual diagnosis technique is exhausting, error-prone, and lengthy, it may result in a patient's death or a higher chance of future disability. In order to counter this problem, some effective automated systems have arisen. These solutions try to recognize stroke medical emergencies automatically and assist clinicians in initiating treatment procedures at the earliest possible stage of stroke onset. As a result, a new approach is presented for detecting a stroke or hemorrhage in CT scan pictures of a patient in this study.

Even though many academics have developed computeraided diagnostic (CAD) systems for stroke, no clinically recognized CAD works with CT brain images as input. Many of these technologies use Machine Learning and specifically Deep Neural Networks to train models that help doctors diagnose strokes faster and more accurately. Gao et al., in 2017, worked with a dataset of brain images from people with Alzheimer's Disease, injuries, and healthy brains. They used 2D Convolutional Neural Networks (CNNs), 2D SIFT, 2D KAZE, 3D SIFT, and 3D KAZE models to classify the images in the three mentioned categories. The average results were 88.8%, 76.7% y 95% for the precision metric in the classification task (Gao, X. W., Hui, R., Tian, Z., 2017). Gautam et al., in 2021, used real brain CT images from

the Himalayan Institute of Medical Sciences (HIMS) to train a model based on image fusion and CNN approaches. By doing so, a mean classification accuracy of 95% was achieved. The proposed model was a 13 layer CNN architecture, and was compared to AlexNet and ResNet50 in the same task (Gautam, A., Raman, B, 2021). These are classifying tools, so it is not possible for doctors to see and evaluate the region where the brain image is abnormal. Some authors have tackled this problem by implementing models that segment the region of the image in which an abnormality can be found. An example of this is the study made by Yahiaoui et al. (2016), in which the brain CT images were enhanced using Laplacian Pyramid (LP). Then the ischemic area was extracted by using the Fuzzy C-Means clustering algorithm.

Given the improvement of accuracy when using CNNs and specifically the U-Net structure for the segmentation of medical images, the purpose of this study is to apply this model to develop the primary function of a CAD tool that helps doctors identify strokes and hemorrhages in CT images of the brain of patients. This function is precisely the ability of the program to segment the region of the brain image in which an injury is present. In order to achieve this objective, several tasks have to be completed. First, it is vital to understand the way CNNs work, including the different operations like convolution, pooling, flatting, deconvolution, and concatenating. Additionally, it is essential to comprehend the structure of the U-Net model and how images are passed layer through layer until the output of a mask that shows the area in which there is a higher chance of finding a stroke, and in consequence, the doctor should pay more attention. Finally, the evaluation of the developed model is crucial for comparing previous and future studies related to brain image segmentation. All these tasks will be discussed in the subsequent sections of this paper

MATERIALS AND METHODS

A. Database

The database that was used for the training of the models consisted of brain images of Computerized Tomography scans that are publicly available. The dataset, which Hssayeni published in 2019, contains 82 CT scans collected between February and August 2018 in the Al Hilla Teaching Hospital, Iraq. Each CT scan included 30 brain image slices with a separation of 5 mm between them. The mean age of the patients was 27.8 years, with a standard deviation was 19.5 years. 46 of the patients were males, and 36 were females. The personal information of each patient was anonymized before the dataset was published. Two radiologists that did not have access to the patient's history determined the presence of an intracranial hemorrhage, the type of intracranial hemorrhage if existing, and the occurrence of a fracture. Both radiologists got to a consensus before emitting the final judgment over each image of the study. Out of the 82 CT scans, 36 were diagnosed with intracranial hemorrhage. The radiologists delineated the intracranial hemorrhage regions in each image if there was an injury. This was saved as a white region in a black 650x650 image. Additionally, brain and bone grayscale 650x650 images were saved for each CT slide (Hssayeni, M., 2019). The dataset included all the information already mentioned. However, since the purpose of the study is to segment the region of the image that contains the intracranial hemorrhage, just the brain window images and the injury masks were used (see Figure 1).

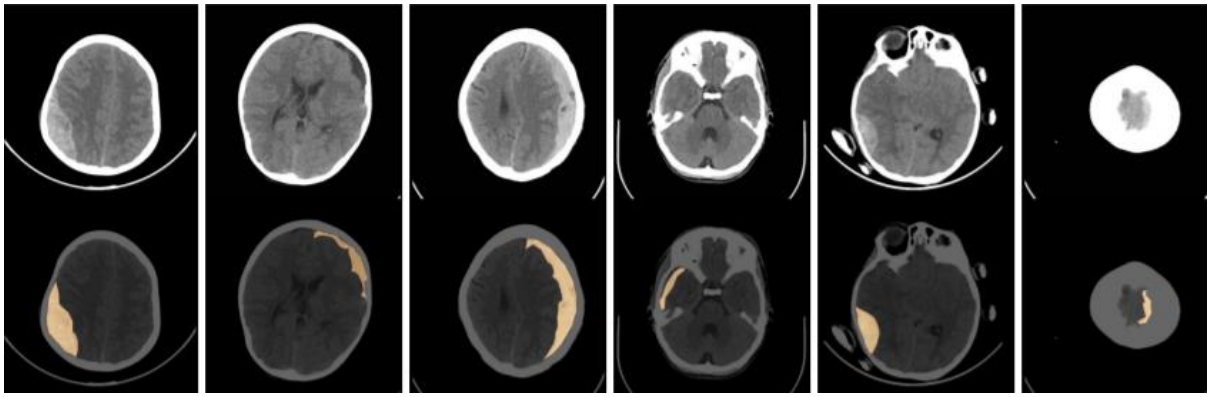

Figure 1: Sample of images in database: original brain CT scan images (first row) and the associated intracerebral hemorrhage ground truth

B. Deep learning models

According to IBM, deep learning is a subset of machine learning that mimics the human brain to attain impressively accurate predictions. Specifically, deep learning models can be defined as neural network models with three or more layers. These models try to imitate the way humans learn and build knowledge by analyzing large volumes of data (IBM Cloud Education, 2020). Some of the advantages of this type of empirical model are its unbiased nature and the ability to process different kinds of information. For example, linear regression is a model with a strong bias because an assumption about the data structure is assumed. On the other hand, deep neural networks are less biased since the training defines the behavior of the model. Additionally, neural networks can process raw, unstructured data as inputs, such as images, audio, and video. This includes the fact that deep neural networks process all this information and determine their relevant features without the intervention of a human experimenter (Yang, C., 2020). These advantages and several more come to a cost. Training these models may require enormous amounts of computational work. Deep learning, despite this disadvantage, has been used more and more frequently due to the use of GPUs in the process of training these models. Clusters of GPUs have been used to decrease the training time by processing the information in a paralleled architecture (Najafabadi, M., Villanustre, F., Khoshgoftaar, T. et al., 2015). The applications of this type of neural network include natural

language processing, scene understanding, character recognition, driverless vehicles operation, medical image processing, marine species recognition, gene expression modeling, gaming, among others (Shinde, P. & Shah, S., 2018).

A subset of deep learning models that are especially recognized for their performance in machine learning tasks is the Convolutional Neural Networks (CNNs) (Saha, S., 2018). This type of network is characterized by layers that implement a linear mathematical operation between matrixes called convolution. CNN models have multiple layers, including convolutional, pooling, fully connected, and non-linearity layers (Albawi, S., Mohammed, T. & Al-Zawi, S., 2017). These layers are combined strategically to obtain the essential data features and fulfill the machine learning tasks. CNNs have had astounding performance in medical image processing, facial recognition, natural language processing, among other applications (Kim, P., 2017). In the image processing field, before the use of these networks, feature extraction was a manual, extensive process that was used to identify objects in the images (Albawi, S., Mohammed, T. & Al-Zawi, S., 2017). Convolutional neural networks solve this problem by automatizing this process and creating a more scalable approach to image classification and object recognition tasks. However, the training stage of these models requires a lot of computational processing time, and for this reason, it is said that CNNs can be computationally demanding. A solution for this issue has been found in using GPUs to train models and reduce the time that this process requires (IBM Cloud Education., 2020).

Among CNNs, several architectures have been created through the years in order to solve different tasks and challenges. One of the very first successful architectures created was Alexnet, created by Alex Krizhevsky, winner of the ImageNet Large Scale Visual Recognition Challenge in 2012 (Wei, J., 2020). Since then, new models have been proposed that have outperformed their predecessors. One of the most important architectures in the medical field

currently is UNET. This model is mainly used for achieving improved performance in image segmentation (Ronneberger, O., Fischer, P., & Brox, T., 2015) and has attained several recognitions, including the victory in the BraTS competition (University of Pennsylvania, 2017). Its architecture consists of three important parts, shaping a letter 'U,' from which it is named is derived. The three sections are contraction, bottleneck, and expansion. The contraction section takes as input the image. Inside this section, each contraction block has two convolutional layers of 3x3 as kernel size, followed by a 2x2 max-pooling layer. The number of filters between each block doubles, so the layer can learn high complexity features from the image in each convolution. The second block consists of two convolutional layers of 3x3, each followed by a max-pooling layer, and at the end of the block, a 2x2 up convolutional layer appears (Lamba, H., 2019). This last layer is essential because it is the one that gives the information to start the last section of the network. Finally, the expansion section is the real heart of this UNET architecture (Sankesara, H., 2019). The expansion section must have as many blocks as the contraction section. However, the slight difference is that each block in the expansion section halves the number of filters to maintain symmetry with the other sections. The most exciting part of this section is that, in each block, the input is also appended with feature maps of its corresponding contraction layer. With this, the network guarantees that the features learned during the contraction steps are also used to reconstruct the image during the expansion step. Each block in this section has convolutional layers of 3x3 and a transposed convolutional layer with a kernel size of 2x2. This layer is responsible for the upsampling process that converts the smaller-sized map to the upper-level map size (double of height and double of width) to be concatenated to the resulting map of the symmetric convolutional block in the contraction path. At the end of the expansion section, a traditional 3x3 convolutional layer appears with the same number of filters as the number of classes desired. The contraction path is responsible for discovering the features or getting information about what is in the

image. However, by doing so, it losses information about the ‘where’ since the image reduction causes a loss of resolution. The purpose of the expansion part is to recover the information of the spatial location of the features, the ‘where,’ to combine this information with the identified feature and segment the image (Lamba, H., 2019). This UNET architecture was used as starting point to personalize our model to solve the initial problem.

C. Proposed method

The traditional U-Net architecture is a great model that has outperformed many other famous CNN models in brain image segmentation competitions, which is the task that the proposed model must solve. However, this architecture does not match the exact requirements of the current study. As mentioned previously, the database used for this study did not have a large number of images for training. This is an issue for the U-Net architecture since it needs numerous images for training to avoid overfitting. This is an issue for many other studies since large datasets are not always available. Taking the original U-Net model as a baseline, two proposed architectures were developed and tested to compare their performance. In terms of architecture, they are very similar. However, they vary on the depth of the model, the dropout layers, and the number of filters that their convolutional blocks used. These models were developed using Python version 3.6.9 and well-known deep learning libraries such as Tensorflow version 1.14.

In order to explain better how both proposed models were created, Figure 2 shows a diagram of the architecture that was defined for one of them. This figure shows that the input of the model is a 256x256 px image of the brain. This image is fed to the first convolutional block. The block comprises a convolutional layer with 64 3x3 filters, an activation layer with a rectified linear function, another convolutional layer with 64 3x3 filters, a batch normalization layer, and an activation layer with a rectified linear function. The job of this

convolutional block is to discover the features that are present in the image by applying the different filters that extract this information. After the convolutional block, a max-pooling layer is added with kernel size 2x2. The purpose of this layer is to choose the most relevant features and create a more specialized map with them. This new map is fed to a new convolutional block, similar to the previous one, but with 128 filters in each convolutional layer. This block is again followed by a max-pooling layer with kernel size 2x2. Another convolutional block is added with 256 filters in each convolutional layer, then a max-pooling layer and a new convolutional block with 512 filters in each convolutional layer. The contraction path and the bottleneck were created until this point, so the next steps correspond to the expansion path. A transposed convolutional layer with a 2x2 kernel and a 2x2 stride was added. This layer has the job of upscaling the map from a deeper level to create a bigger map that can be concatenated with the resulting map of the convolutional block in the symmetrical contraction path level. This ensures that the feature information is selected and combined with the spatial information of the upper-level map (Anwar, A., 2021). After the concatenation, a convolutional block was added with 256 filters in each convolutional layer. Then, another transposed convolutional layer was added, and the output was concatenated with the resulting map of the symmetrical convolutional block of the contraction path. A convolutional block was added, with 128 filters in each convolutional layer. The upsampling, concatenation, and convolution (with 64 filters) were repeated once again, and the output was fed to a convolutional layer with one filter that corresponded to the prediction of the model. This final layer had a 1x1 kernel and a sigmoid activation function. This function allows the model to output a value between 0 and 1 for each pixel in the resulting map or mask.

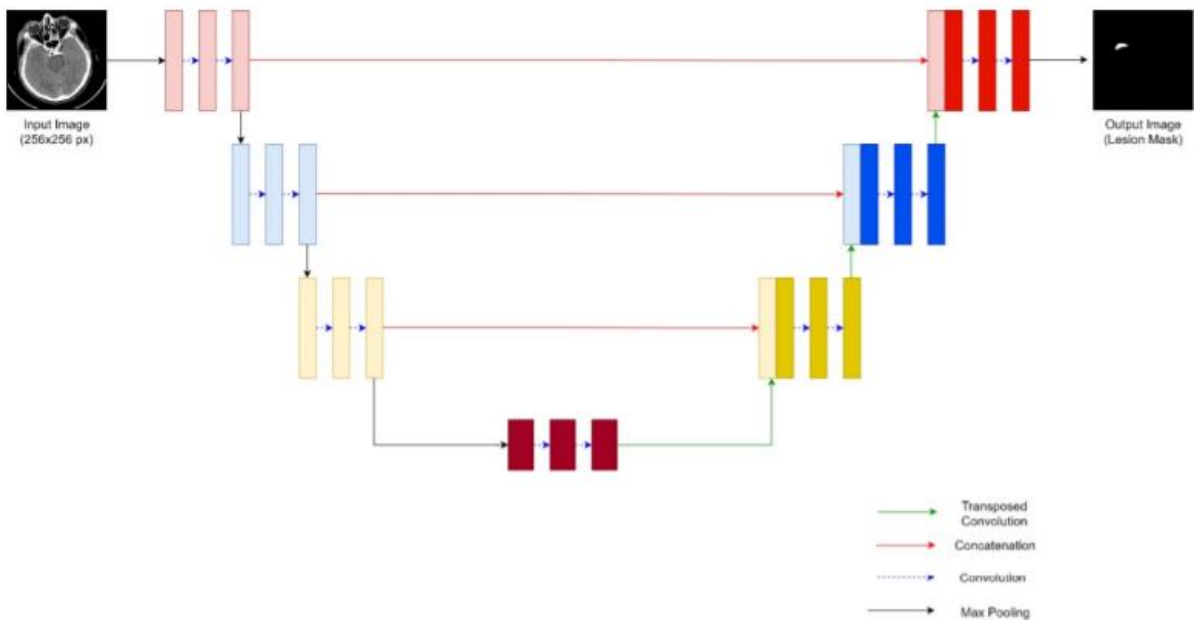

Figure 2: Diagram of the first proposed model that is based on the U-Net architecture.

The second proposed model included additional Dropout layers to avoid the overfitting and the vanishing gradient problem (Tan, H., & Lim, K., 2019). These are issues that deep neural networks models that use gradient-based backpropagation suffer. The Dropout layer randomly changes the input and sets it to 0, depending on the frequency defined as a parameter. This helps to prevent overfitting by adding variation to the model (Keras Team, 2021). The architecture described for the first proposed model was modified by adding these dropout layers in the convolutional block. Specifically, a dropout layer with a frequency rate of 0.1 was added between the convolutional layers in the same convolutional block. This was done to see the effect of the inclusion of these layers in the solution of the overfitting problem.

Concerning the output of the model, it is a map with the same size as the input image (256x256 px) that represents the region of the image in which the model predicts there is an injury. This predicted map was compared with the original mask that the radiologists created by manually delineating the region where they identified the injury. Each of the pixels in the predicted map has a probability of being part of the region of the injury. The output has the

exact size of the input image since its purpose is to superimpose the predicted mask over an image that a radiologist wants to analyze with this program.

D. Experimental setup

The experimental setup was organized in several stages to explain this process.

1) *Data preprocessing:* As explained in the previous dataset section, the original database consisted of CT scans of 82 patients, with an average of 30 brain images for each scan. The total amount of images was 2500. However, just 318 of these images contained an injury. This evidenced that the database was unbalanced. To solve this issue, all the images that contained a proven injury were selected along the same amount of randomly selected images from those that did not have an injury. This was done to balance the dataset and avoid biasing the models in their segmentation task. This process left a dataset of 636 images. The dataset balancing created a new issue. Deep learning models tend to need many data for the training process, so 636 images may not have been enough to train the models adequately. For this reason, a process of data augmentation was added. The process that was included for the data augmentation was the reflection of the image over the y axis so that the image would be flipped horizontally. The reflection over the x-axis was also included to flip the image vertically. It should be mentioned that different image enhancement methods were also tried, for example, CLAHE histogram equalization. However, when the image was visually evaluated, it could be seen that the lesions were more easily confused with the bone tissue in the image. For this reason, it was decided not to use histogram equalization even though it is a common practice in the processing of other medical images. Finally, all the images were resized to a smaller dimension. This was done to

decrease the time and memory required for training. The original dimension of the images was 650x650 pixels and was resized to 256x256 pixels.

2) *Training and test sets:* The dataset was separated into training, validation, and test sets. The proportions of the data used for each set were 80% for training, 10% for validation, and 10% for testing. This was done to have as much data as possible for training and so that the model could perform adequately. Additionally, it should be mentioned that although there are indeed other recommended methods for partitioning the data, such as the k-fold cross-validation, this type of procedure requires training numerous models prior to obtaining the metrics. Since the proposed model has a deep architecture, the training time can be extended considerably and thus making the training step more timeconsuming. Considering this, it was decided to use a stratified partition method that would maintain the proportion of images with lesions and without lesions within each set.

3) *Model configuration:* For both proposed models, some of the hyperparameters were modified in order to fine tune the models with these parameters and obtain the best results. The hyperparameters that were varied specifically for the experiments were the epochs for which the model was trained, and the batch size that was used. Regarding the epochs, the models were trained in a range of 200 to 1,400 epochs with steps of 200. This was done in order to determine the point at which overfitting began to occur in the data, if any existed, or if at some point the model no longer improved considerably despite training it with even more epochs. The models were trained also varying the batch sizes in two levels: 16 and 32. The optimizer that was used was Adam, which is one of the most popular optimizers due to its computational efficiency, little memory requirement, and adequate performance with large datasets. It is important to mention that this optimizer receives as a parameter a learning rate that was set to 0.0001;

however, due to its operation, this optimization algorithm varies the learning rate so that training is more efficient (Kingma, D., & Ba, J., 2017). Moreover, the dropout layers that were added to the second model had a set frequency rate of 0.1 in order to reduce the overfitting. This affects the model only during the training phase. Each dropout layer deactivates the input of the following layer based on the defined frequency. In this way, some randomness is added to the model, and thus, overfitting is avoided.

4) *Assessment metrics:* Several metrics were used to evaluate and compare the models' performance. The metrics most commonly used in neural networks evaluation are the area under the ROC curve, precision, recall, the F1- score, and accuracy. However, the segmentation task requires slightly different metrics to communicate the performance more objectively and straightforwardly. For this study, the first metric that was used was accuracy. Since it is a segmentation task, accuracy can also be understood as pixel accuracy. This metric represents the proportion of pixels in the prediction classified correctly according to the ground truth mask. This metric is too loose and biased because the model can obtain very high scores almost without segmenting the image correctly. [24] For instance, let us consider a ground truth mask in which the region with a lesion represents 5% of the total amount of pixels. If the model predicts a mask with no injury, the pixel accuracy metric would be 95%. Even though this is not a very efficient metric for the evaluation of the model, it is straightforward to understand, and that was why it was kept as one of the metrics for this study. Nonetheless, additional metrics were defined for this study, and the loss component was bound to one of these metrics.

First, the Jaccard distance was calculated, also called Intersection over Union. This metric corresponds to the intersection or area of overlap between the predicted

mask and the ground truth mask, divided by the area of the union of both masks (Wang, Z., Wang, E. & Zhu, Y., 2020). The IoU was calculated in the following way. The intersection was calculated by summing the product of each pixel in the ground truth mask and the pixel with the exact coordinates in the predicted mask. The union was calculated by summing the number of pixels that the ground truth had classified as lesion and the number of pixels that the prediction had classified as a lesion and subtracting the intersection. This was also done for the case in which both the ground truth and the prediction had a pixel classified as background, and then the result was averaged. This metric can take values from 0 to 1 (0 to 100%) with 0, meaning that the model predicted precisely the opposite of the ground truth (the background as the lesion and vice versa) and 1 meaning that the prediction is the same as the ground truth (Tiu, E., 2020).

Another metric that was defined was the Dice Coefficient. This metric is very similar to the Jaccard distance; however, this metric was used to develop a loss function that the optimizer would try to minimize and thus make the model performs better. The Dice Coefficient, also known as the F1 score, is equal to two times the area of overlap of the masks divided by the sum of the total pixels in both masks (Wang, Z., Wang, E. & Zhu, Y., 2020). This metric can also take values from 0 to 1, so to make this a loss function, this value was multiplied by -1. In this way, the Adam optimizer would try to minimize this negative value and thus maximize the overlap between the predicted and the ground truth masks.

RESULTS AND DISCUSSION

The experiments were run using an NVidia DGX workstation using a GPU for faster training of the models. Despite this, it is important to note that, on average, it took a day of processing to run 300 epochs of the process. The results are shown and discussed in the next section.

A. Performance evaluation

Table I shows the performance results for the two models developed, trained, and used for experimentation, including the different levels of the factors modified in each run. Epochs were variated from 200 to 1000 in 200 steps, and batch sizes of 16 and 32 were used. For the first model, it can be seen that the pixel accuracy metric is, as it was mentioned earlier, very high even in the first experiments with fewer training epochs. This is because of the existing bias, so that we will focus the discussion on the other metrics. For instance, it is interesting to notice the evolution of the IoU score. Let us suppose that there is a 10% region of the image that contains a lesion. However, if the model only predicts an empty mask (indicating no injury), the IoU would be 0.45. This is because it has intersected correctly 90% of the pixels without an injury and 0% of the pixels with an injury, giving an average of 45% or 0.45. It was taking this idea in mind. In its lower point, with 200 epochs and a batch size of 16, model 1 obtains an IoU of 0.56. This is a relatively low value since it conveys that the model predicts a mask that intersects the ground truth mask only in a 56%. However, as the model is trained for more epochs, the IoU improves considerably. The best combination of hyperparameters was attained when model 1 was trained for 1000 epochs with a batch size of 16. The IoU during the model testing was 0.79, and the Dice Coefficient was 0.83. It is important to notice that the evolution of the model from 800 epochs to 1000 epochs is just three percentual points in the IoU and one percentual point in the Dice Coefficient, showing therefore that the model is not improving as

much in the first experiments, since the evolution from 200 epochs to 400 epochs is 12 and 13 percentual points in the same metrics. Moreover, the maximum performance results are improvable since it can be understood that, on average, 80% of the predicted mask intersects correctly with the ground truth.

The second model has better performance results in all the metrics and better generalization power. Since the first experiments with 200 epochs, the model presented slightly better results than the previous one. This tendency continued for all the experiments. The best results of this model showed that this architecture outperformed the first one by several percentual points in both metrics, obtaining an intersection over union metric of 0.85 and a dice coefficient metric of 0.89. These are excellent results for a segmentation model because it indicates that a significant part of the lesion has been detected. The difference between the second model and the first one was the presence of dropout layers. These layers alter the model's behavior during the training phase by randomly deactivating some of the layer's inputs that go after the dropout layer. This is done to avoid overfitting and give the model a better generalization power. By having an additional random factor, the model does not memorize the same patterns to segment. However, it learns in a more general way. This is shown in this case because the results obtained by the second model in the testing phase were substantially better than the ones attained by the first model that did not include these layers.

Moreover, Figure 3 shows the model's behavior during the training and validation phases across the different epochs. This graph shows the learning process and its effect on the evolution of the intersection over union metric and the loss. The training curves show that the model learns correctly to develop the task of segmenting the image since the IoU increases in the first epochs very fast and then progressively slower. However, it reaches an asymptote that could be defined as the maximum that this model can attain.

Table 1: Performance results of the deep learning proposed models with different hyperparameters.

Arch.	Conv. Layer (f)	Kernel Size	Pool size per Layer	Dropout Layers	Batch Size	Epochs (u)	IoU (u)	Dice Cf.	ACC (%)
Model 1	(64,128,256,512)	(3x3)	(2x2)	Not included	16	200	0.56	0.65	99.74
					32	200	0.57	0.65	99.75
					16	400	0.68	0.78	99.77
					32	400	0.67	0.77	99.77
					16	600	0.72	0.80	99.78
					32	600	0.73	0.81	99.78
					16	800	0.76	0.82	99.81
					32	800	0.77	0.82	99.80
					16	**1000**	**0.79**	**0.83**	**99.82**
					32	1000	0.78	0.83	99.82
Model 2	(64,128,256,512)	(3x3)	(2x2)	Yes, between the conv. layers of each conv. block	16	200	0.60	0.69	99.73
					32	200	0.61	0.70	99.75
					16	400	0.72	0.81	99.82
					32	400	0.72	0.81	99.82
					16	600	0.80	0.85	99.87
					32	600	0.79	0.85	99.86
					16	800	0.84	0.88	99.90
					32	800	0.84	0.88	99.90
					16	1000	0.84	0.88	99.90
					32	**1000**	**0.85**	**0.89**	**99.91**

Conv.- convolutional; f- number of filters per layer; u- units; IoU- Intersection over Union metric; Cf- coefficient; ACC- pixel accuracy metric.

The loss curve is very similar, with the difference that it decreases until it stabilizes and does not decrease substantially even though more training epochs pass. On the other hand, the behavior of the validation curves can be seen as more unstable, with a broader range of variation; however, it can be identified the same tendency of increasing (IoU) or deceasing (loss) until a certain level in which the rest of variations are explicable due to the difference of the data. CT scan images are a type of data that can be different. The image has a grayscale, making it harder for the model to segment the lesion correctly because it can not use a change of color as a feature. Additionally, since there are several slices in a CT scan, the amount of bone tissue and other organs that appear in a tomography adds variability to the images. These could be the reasons behind the unstable behavior of the validation curves. Despite this, it could be said that the model learns correctly and validly fulfills its task.

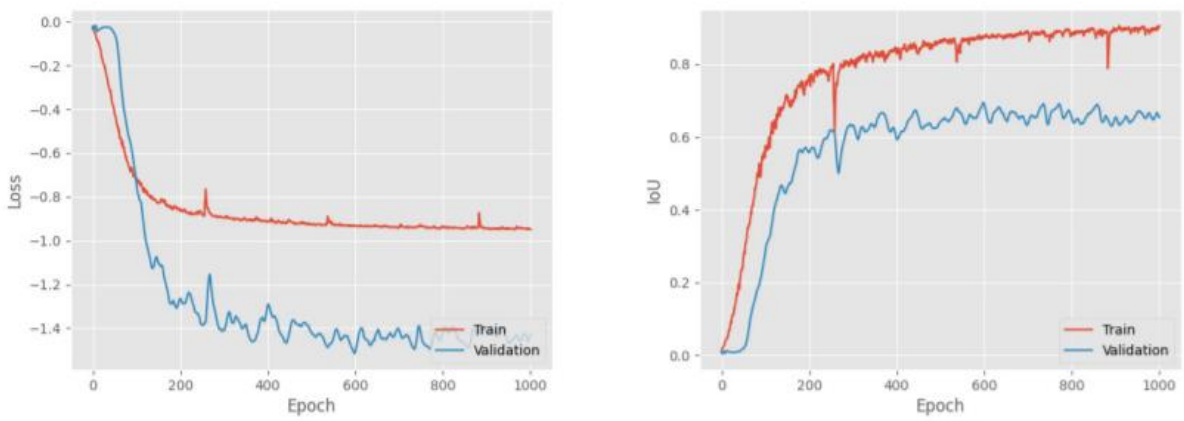

Figure 3: Performance of the best proposed deep learning model (model 1) on segmentation task during the training and the validation stages

Finally, a table with some examples of the lesions and the comparison of the ground truth segmentation versus the model prediction segmentation is presented in Figure 4. This image shows that the model has a great capacity to segment the image and identify the region of the image where a lesion is located. It is interesting to see that the model can work with lesions that are present as a single mass together but of correctly segmenting different regions within

the same image. Similarly, it can be shown that segmentation has an excellent ability to determine the correct shape of the region in which the lesion is located. This is a crucial difference with the models with bounding boxes and ROIs. In this way, the neural network is already in charge of identifying the contour of the lesion, which can provide more valuable information for the doctors who read the images that this model processes. Finally, it is worth mentioning that the mask predicted by the best-proposed model works with probabilities that are graphed on a scale that shows areas that are close to 0.5 in white. Progressively it assigns a more intense red color as it approaches 1. This is important as only the edges of the lesion have a white color, which is correct because getting closer to the edge makes it more difficult to distinguish between the lesion and the normal tissue in the image. However, there are no white dots within the segmented area, which also shows more of the desired capabilities of the model.

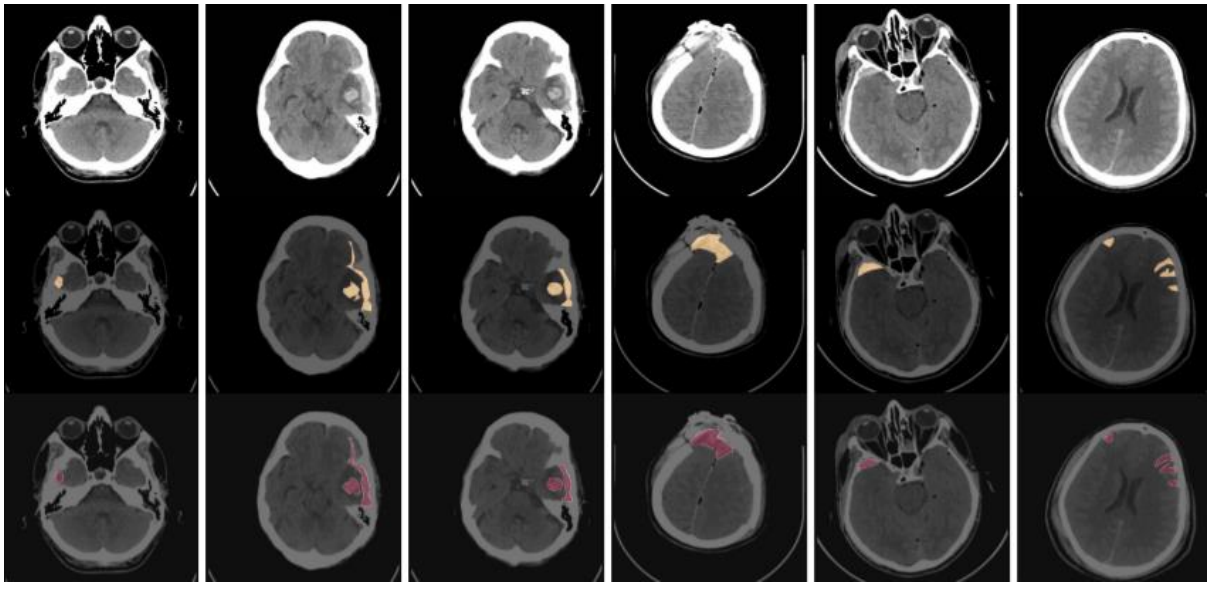

Figure 4: Examples of successful segmentation performance of the best model. The original CT scan images (first row), lesion ground truth

CONCLUSIONS AND FUTURE WORK

The purpose of this research was to propose several models that could be used as the heart of a CAD tool. This was done by defining two different CNN architectures based on a wellknown U-Net model. In the end, the second proposed model was the one that performed better, and thus the selected model at the end of this study. It is essential to mention that the task of this model was to segment the image and show through a mask the region in which the image presented a lesion. For this reason, the metrics that the model used were slightly different, and excellent results were achieved in the intersection over union and the dice coefficient by the best model architecture that was trained for 1000 epochs using a batch size of 32. This model attained a 0.85 value for the IoU and a 0.89 for the dice coefficient.

One of the most important things that could be noticed with this study is that the dropout layers included in the second model gave it a better generalizing power that the first model because of the improvement in the performance results in the testing phase. For the testing, 10% of the total images were separated, and the model did not contact these images during the validation and the training phases. The fact that the second model had better results with these images shows that the dropout layers fulfilled their purpose during the training phase by avoiding the overfitting of the model. This architecture could segment in a better manner the lesion region in these unknown images, and the results were excellent. The only other difference with the best configuration of the first model was the batch size. However, as it was seen in Table 1, the results for the same model using the same number of epochs but different batch sizes were very similar. For this reason, it is not considered a relevant factor. However, as a further work proposal, the extension of this conclusion could be expanded by implementing a different partition of the data. With more runs, more statistical data could be used in ANOVA or means difference tests in order to determine the

relevance of each of the factors that were manipulated in the study (batch size and epochs) or additional factors such as dropout frequency rates, number of filters, optimizers and size of the kernels and strides in the convolutional and max-pooling layers.

Finally, as shown in Figure 4, the selected model performs adequately and segments the region of the image in which there is a lesion. Moreover, the model returned empty masks correctly for all the images that did not present any lesion. From this, it could be inferred that the model performs very well in a classification task; however, no formal procedure was defined so that the model could return a classification tag in which it mentioned if the image contained a lesion and determined the type of intracranial hemorrhage that was each one of the lesions. For this reason, a proposal for future work is to implement another procedure, module, or part of the model that uses the identified features and return information about the typology of the lesions and the presence or absence of them in the image.

ACKNOWLEDGEMENT

Authors thank the Applied Signal Processing and Machine Learning Research Group of USFQ for providing the computing infrastructure (NVidia DGX workstation) to implement and execute the developed source code.

REFERENCES

Albawi, S., Mohammed, T. & Al-Zawi, S. (2017). *Understanding of a convolutional neural network.* International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.

Anwar, A. (2021, April 16). *What is transposed convolutional layer?* Medium. Retrieved September 15, 2021, from https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11

Centers for Disease Control and Prevention. (2021, May 25). *Stroke facts*. Centers for Disease Control and Prevention. Retrieved September 28, 2021, from https://www.cdc.gov/stroke/facts.htm

Gao, X. W., Hui, R., & Tian, Z. (2017). *Classification of CT Brain Images based on Deep Learning Networks.* Computer Methods and Programs in Biomedicine, 138, 49–56. https://doi.org/10.1016/j.cmpb.2016.10.007

Gautam, A., & Raman, B. (2021). *Towards effective classification of brain hemorrhagic and ischemic stroke using CNN.* Biomedical Signal Processing and Control, 63, 102178. https://doi.org/10.1016/j.bspc.2020.102178

Harpaz, D., Eltzov, E., Seet, R. et al. (2017). *Point-of-Care-Testing in Acute Stroke Management: An Unmet Need Ripe for Technological Harvest*. Biosensors, 7(3), 30. https://doi.org/10.3390/bios7030030

Hssayeni, M. (2019, August). *Computed tomography images for intracranial hemorrhage detection and segmentation*. PhysioNet 2019.

IBM Cloud Education. (2020, May 1). *What is deep learning?* IBM. Retrieved August 28, 2021, from https://www.ibm.com/cloud/learn/deep-learning#toc-deep-learn-g0Ru2Ceu

IBM Cloud Education. (2020, October 20). What are convolutional neural networks? IBM. Retrieved November 10, 2021, from https://www.ibm.com/cloud/learn/convolutional-neural-networks

Keras Team. (2021). *Keras documentation: Dropout layer*. Keras. Retrieved August 24, 2021, from https://keras.io/api/layers/regularization_layers/dropout/

Kim, P. (2017). *Convolutional Neural Network*. In: MATLAB Deep Learning. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2845-6_6

Kingma, D., & Ba, J. (2017). *Adam: A method for stochastic optimization.*

Lamba, H. (2019, February 17). *Understanding semantic segmentation with UNET*. Medium. Retrieved October 28, 2021, from https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47

Najafabadi, M., Villanustre, F., Khoshgoftaar, T. et al. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data 2, 1. https://doi.org/10.1186/s40537-014-0007-7

Ronneberger, O., Fischer, P., & Brox, T. (2015, May 18). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv. Retrieved October 28, 2021, from https://arxiv.org/abs/1505.04597

Saha, S. (2018, December 17). *A comprehensive guide to Convolutional Neural Networks - the eli5 way*. Medium. Retrieved November 17, 2021, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Sankesara, H. (2019, January 23). *U-Net*. Medium. Retrieved November 20, 2021, from https://towardsdatascience.com/u-net-b229b32b4a71

Shinde, P. & Shah, S. (2018). *A review of machine learning and deep learning applications*. in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1–6.

Tan, H., & Lim, K. (2019). *Vanishing gradient mitigation with deep learning neural network optimization.* 7th International Conference on Smart Computing Communications (ICSCC), 2019, pp. 1–4

Tiu, E. (2020, October 3). *Metrics to evaluate your semantic segmentation model*. Medium. Retrieved December 10, 2021, from https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

University of Pennsylvania. (2017). *Section for biomedical image analysis (SBIA)*. MICCAI BraTS 2017: Scope Section for Biomedical Image Analysis (SBIA), Perelman School of Medicine at the University of Pennsylvania. Retrieved December 1, 2021, from https://www.med.upenn.edu/sbia/brats2017.html

Wang, Z., Wang, E. & Zhu, Y. (2020). *Image segmentation evaluation: a survey of methods.* Artif Intell Rev 53, 5637–5674 (2020). https://doi.org/10.1007/s10462-020-09830-9

Wei, J. (2020, September 25). *Alexnet: The architecture that challenged CNNS*. Medium. Retrieved September 5, 2021, from https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951

Yahiaoui, A. & Bessaid, A. (2016). *Segmentation of ischemic stroke area from ct brain images*. 2016 International Symposium on Signal, Image, Video and Communications (ISIVC), pp. 13–17.

Yang, C. (2020, February 27). *Deep learning in science*. Medium. Retrieved October 20, 2021, from https://towardsdatascience.com/deep-learning-in-science-fd614bb3f3ce