

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií  
FIIT-100241-91983

Ahmed Lotfi Alqnatri

Spracovanie obrazových medicínskych dát metódami počítačového  
videnia a hlbokých neurónových sietí

## **Bakalárska práca**

Študijný program: Informatika

Študijný odbor: Informatika

Miesto vypracovania: UPAI

Vedúci práce: doc. Ing. Vanda Benešová, PhD.

Máj 2021



Slovak University of Technology in Bratislava  
Faculty of Informatics and Information Technologies  
FIIT-100241-91983

Ahmed Lotfi Alqnatri

Processing of medical image data by computer vision methods and deep  
neural networks

## **Bachelor project**

Study program: Informatics

Field: Informatics

Place of elaboration: UPAI

Supervisor: doc. Ing. Vanda Benešová, PhD.

May 2021





## BACHELOR THESIS TOPIC

Student: **Ahmed Lotfi Alqnatri**  
Student's ID: 91983  
Study programme: Informatics (conversion programme with a foundation year)  
Study field: Computer Science  
Thesis supervisor: doc. Ing. Vanda Benešová, PhD.

Topic: **Spracovanie obrazových medicínskych dát metódami počítačového videnia a hlbokých neurónových sietí**

Language of thesis: English

Specification of Assignment:

Pri spracovaní diagnostických dát v medicíne, ako sú napríklad dáta z počítačovej tomografie alebo magnetickej rezonancie, nachádzajú metódy počítačového videnia stále významnejšie uplatnenie. Významnými problémovými oblasťami sú hlavne detekcia a segmentácia anomálií a registrácia multimodálnych dát. Analyzujte súčasný stav problematiky využitia počítačového videnia pri spracovaní vizuálnych dát v medicíne. Zamerajte sa predovšetkým na metódy využívajúce hlboké neurónové siete. Navrhnite metódu na podporu diagnostiky s využitím moderných metód umelej inteligencie. Navrhnutú metódu realizujte softvérovým prototypom s využitím knižnic a rámcov vhodných na spracovanie medicínskych dát. Riešenie overte experimentom s reálnymi dátami z medicínskych modalít. Vyhodnoťte presnosť, robustnosť a časovú efektívnosť spracovania. Výsledky porovnajte s inými publikovanými riešeniami.

Length of thesis: 40

Assignment procedure from: 17. 02. 2020

Date of thesis submission: 17. 05. 2021

**Ahmed Lotfi Alqnatri**  
Student

**Ing. Katarína Jelemenská, PhD.**  
Head of department

**doc. Ing. Valentino Vranič, PhD.**  
Study programme supervisor





Čestne vyhlasujem, že som túto prácu vypracoval samostatne, na základe konzultácií a s použitím uvedenej literatúry.

V Bratislave, 17.05.2021

Ahmed Lotfi Alqnatri





## **Pod'akovanie**

Ďakujem vedcovi bakalárskej práce doc. Ing. Vanda Benešová, PhD. a pedagogickým vedám za cenné rady a pomoc poskytnuté pri spracovaní bakalárskej práce. Špeciálne pod'akovanie patrí mojej rodine, priateľom a učiteľom.



# Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Informatika

Autor: Ahmed Lotfi Alqnatri

Bakalárska práca: Spracovanie obrazových medicínskych dát metódami počítačového videnia a hlbokých neurónových sietí

Vedúci bakalárskej práce: doc. Ing. Vanda Benešová, PhD.

Máj 2021

Prístupy počítačového videnia sa stávajú obzvlášť bežnými pri zbere lekárskeho diagnostických údajov, ako sú údaje z počítačovej tomografie alebo magnetickej rezonancie. Identifikácia a segmentácia nezrovnalostí, ako aj registrácia multimodálnych údajov sú hlavnými problémovými oblasťami. V tejto bakalárskej práci sme teda chceli vyriešiť typický problém, ktorý sa každodenne vyskytuje na pohotovosti, a to intrakraniálne krvácanie, ktoré je smrteľné, ak nie je zistené dostatočne rýchlo na to, aby sme ho pochopili a skontrolovali údaje, ktoré nám boli poskytnuté, vyvinuli sme algoritmus hlbokého učenia s klasifikačným modelom na detekciu ICH a určenie toho, ktorý podtyp krvácania je na danom obrázku.



# **Annotation**

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree course: Informatics

Author: Ahmed Lotfi Alqnatri

Bachelor's Thesis: Processing of medical image data by computer vision methods and deep neural networks

Supervisor: doc. Ing. Vanda Benešová, PhD.

May 2021

Computer vision approaches are becoming particularly common in medical diagnostic data collection, such as computed tomography or magnetic resonance imaging data. The identification and segmentation of irregularities, as well as the registration of multimodal data, are major problem areas. So, in this bachelor thesis, we wanted to tackle a typical problem seen every day in the emergency room, which is intracranial hemorrhage, which is fatal if it is not detected quickly enough to operate, by understanding the problem and reviewing the data we were given. We developed a deep learning algorithm with a classification model to detect ICH and determine which hemorrhage subtype is present in a given image.



# Contents

<b>1. Introduction .....</b>	<b>1</b>
<b>2. Deep Learning.....</b>	<b>5</b>
<b>3. Convolutional Networks .....</b>	<b>7</b>
3.1. Convolution Layer .....	7
3.2. Pooling Layer .....	12
3.3. Fully Connected Layer .....	14
<b>4. Transfer Learning .....</b>	<b>15</b>
4.1. Overview .....	15
4.2. Approaches to Transfer Learning .....	15
<b>5. Related Work.....</b>	<b>18</b>
5.1. Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning [9].....	18
5.1.1. Methods and Materials .....	19
5.1.2. Evaluation of model performance .....	22
5.2. Precise diagnosis of intracranial hemorrhage and subtypes using a three-dimensional joint convolutional and recurrent neural network [34].....	23
5.2.1. Methods and Materials .....	23
5.2.2. Training and evaluating models .....	24
5.3. Convolutional neural networks for detection intracranial hemorrhage in CT images [29]	26
5.3.1. Methods and Materials .....	26
5.3.2. Convolutional Neural Network .....	27
5.3.3. Training and evaluating models .....	29
<b>6. Our solution .....</b>	<b>30</b>
6.1. Dataset.....	30
6.2. First experiment .....	33
6.2.1. Image Preprocessing .....	33
6.2.2. Model Architecture .....	33



6.2.3.	Training and Validation .....	34
6.3.	Second Experiment .....	35
6.3.1.	Image preprocessing .....	35
6.3.2.	Balancing Data .....	37
6.3.3.	Using Transfer Learning for Classification Task .....	38
6.3.4.	Training and Validation Results.....	42
7.	<i>Conclusion</i> .....	<b>47</b>
8.	<i>Resumé</i> .....	<b>48</b>
9.	<i>Resources</i> .....	<b>53</b>
<i>Appendix A: Work's Progress Plan</i>		
<i>Appendix B: Technical Documentation</i>		
<i>Appendix C: Description Of The Digital Part Of The Work</i>		

# 1. Introduction

## 1.1. Purpose

Computer vision methods are becoming increasingly important in the processing of diagnostic data in medicine, such as computed tomography or magnetic resonance imaging data. Significant problem areas are mainly the detection and segmentation of anomalies and the registration of multimodal data.

Analyze the current state of the issue of the use of computer vision in the processing of visual data in medicine. Focus mainly on methods using deep neural networks. Design a method to support diagnostics using modern methods of artificial intelligence. Implement the proposed method with a software prototype using libraries and frameworks suitable for medical data processing. Verify the solution by experimenting with real data from medical modalities. Evaluate the accuracy, robustness, and time efficiency of processing. Compare the results with other published solutions.

## 1.2. Motivation

Intracranial hemorrhage (ICH) corresponds to bleeding inside the skull caused by a vascular rupture. Speed of diagnosis is crucial because the mortality reaches up to 60% after 30 days and 35% to 52% of patients die before a month after being diagnosed, and approximately half of these deaths occur within the first 24 hours [30].

This is a reason why ICH is considered a medical emergency and specialists must diagnose it properly and quickly. However, up to 20% of patients with reported ICH can be misdiagnosed in general medicine settings and emergency departments, indicating the bleeding cannot be adequately distinguished without the use of medical imaging techniques.

Worldwide, computed tomography (CT) of the brain is used to detect neurologic emergencies. However, expert interpretation of these scans is expected, and even highly qualified experts can ignore subtle life-threatening findings. A special difficulty for head

CT is identifying minor slight anomalies on a multi slice cross-sectional image with maximum or near-perfect sensitivity and very high precision. A special difficulty for head CT is identifying minor subtle anomalies with great or near-perfect sensitivity and very high precision on a multi slice cross-sectional (three-dimensional [3D]) imaging modality marked by poor soft tissue contrast, reduced signal-to-noise using existing low radiation-dose protocols, and a high frequency of artifacts. But by training a fully convolutional neural network we can decrease this margin of human error and show more accurate and relatively fast results. But to implement our solution we must define the problem and its parameters.

### **1.3. Intracranial Hemorrhage**

The term "intracranial hemorrhage" refers to bleeding within the brain. Intracranial hemorrhage is a leading cause of death and injury, as well as a contributing factor to stroke [1]. Intracranial hemorrhage may happen on its own or as a result of trauma [2-4]. Spontaneous intracranial hemorrhage can be caused by several diseases, including but not limited to arteriovenous malformations, ruptured aneurysms, anticoagulation, cancers, venous sinus thrombosis, hypertension, cerebral amyloid angiopathy, and hemorrhagic stroke transfer [1]. Traumatic intracranial hemorrhage may happen to anybody in a trauma situation, but patients on anticoagulants are at a slightly higher risk of intracranial hemorrhage [3]. Intracranial hemorrhage is a medical emergency, and prompt diagnosis is critical for optimizing patient conditions due to rapid patient deterioration within the first few hours of the onset of symptoms [2].

The terms epidural, subdural, subarachnoid, and intraparenchymal refer to the site of the bleeding. Unenhanced computed tomography (CT) scans of the brain are often used to detect intracranial hemorrhage [2]. Differences in x-ray attenuation and intracranial hemorrhage position on unenhanced CT scans of the brain make them detectable and cause the various forms of intracranial hemorrhage to be distinguished. Hemorrhages can be subtle for a variety of causes, including small blood pooling around the edge of typical intracranial tissues with equal attenuation, and the falx cerebri or hemorrhage age of days

or weeks, in which case the attenuation on CT becomes more similar to that of brain tissue. While intracranial hemorrhages are usually hyperattenuating as compared to gray matter, they may also be hypoattenuating. Other brain structures, such as calcifications of the pineal glands and choroid plexus, may exhibit attenuation similar to intracranial hemorrhages.


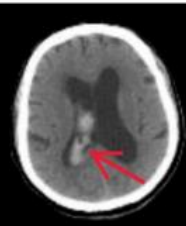
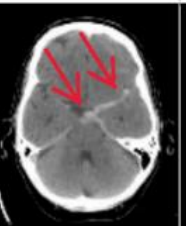
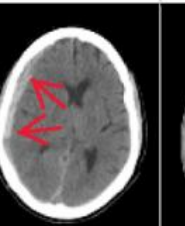
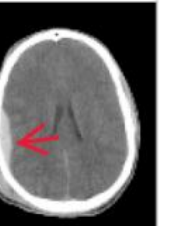
	Intraparenchymal	Intraventricular	Subarachnoid	Subdural	Epidural
Location	Inside of the brain	Inside of the ventricle	Between the arachnoid and the pia mater	Between the Dura and the arachnoid	Between the dura and the skull
Imaging					
Mechanism	High blood pressure, trauma, arteriovenous malformation, tumor, etc	Can be associated with both intraparenchymal and subarachnoid hemorrhages	Rupture of aneurysms or arteriovenous malformations or trauma	Trauma	Trauma or after surgery
Source	Arterial or venous	Arterial or venous	Predominantly arterial	Venous (bridging veins)	Arterial
Shape	Typically rounded	Conforms to ventricular shape	Tracks along the sulci and fissures	Crescent	Lentiform
Presentation	Acute (sudden onset of headache, nausea, vomiting)	Acute (sudden onset of headache, nausea, vomiting)	Acute (worst headache of life)	May be insidious (worsening headache)	Acute (skull fracture and altered mental status)

Figure 1. ICH Subtypes [10]

### 1.3.1. Epidural hematoma

A hematoma is a mass of blood in the form of a clot or ball that forms outside of a blood vessel. When blood accumulates between your scalp and the outermost lining of your brain, it forms an epidural hematoma.

It normally occurs after a head injury, most commonly with a skull fracture. High-pressure bleeding is a prominent feature. If you have an epidural hematoma, you can lose consciousness for a short period of time before regaining consciousness.

### **1.3.2. Subdural hematoma**

A subdural hematoma is a pool of blood on the brain's surface. It is usually caused by the head going quickly forward and then slowing, like with a car crash. However, it may also indicate child neglect. This is the same kind of movement that an infant feels when he or she is shook.

In older adults and people with a history of heavy alcohol consumption, subdural hematomas are more frequent than other ICHs.

### **1.3.3. Intracerebral hemorrhage**

Intracerebral hemorrhage occurs where there is bleeding inside the brain. This is the most frequent kind of ICH caused by a stroke. It is not necessarily caused by an accident.

The abrupt onset of neurological deficit is a common warning sign. This is a concern with the way your brain works. The signs develop over a period of minutes to hours. They are as follows:

Headache, difficulty speaking, nausea, vomiting, decreased consciousness, weakness in one part of the body, elevated blood pressure.

## 2. Deep Learning

A lot of deep learning algorithms have been developed to address unsupervised learning issues, but none of them have fully solved the problem in the same manner as deep learning has effectively solved the supervised learning problem for a wide range of tasks. [20]

Deep learning is a type of machine learning algorithm that employs several layers to extract higher-level features from raw input. Lower layers of image processing, for example, may define edges, while higher layers may identify human-specific concepts such as numbers, letters, or faces. [21]

The majority of current deep learning models are built on artificial neural networks, especially convolutional neural networks (CNNs), but they can also include declarative formulas or explanatory variables ordered layer-wise such as nodes in deep conviction networks and deep Boltzmann machines which are examples of deep generative structures. [22]

Each stage of deep learning learns to translate the input data into a somewhat more abstract and composite representation. The raw data in an image recognition algorithm could be a matrix of pixels, the first representational layer could abstract the pixels and encode borders, the second layer could write and encode edge arrangements, and the third layer could encode a nose and eyes, and the fourth layer can detect the presence of a face in the picture.

Notably, a deep learning method will figure out which features belong in which level all by itself. (Of course, this does not eliminate the need for hand-tuning; changing the number of layers and the amount of the layers, for e.g., would have differing degrees of abstract concept.) [23]

The term "deep" refers to the amount of layers by which the data is transformed. Deep learning models, in particular, have a significant credit assignment path (CAP) depth. The CAP is the series of changes that take place from input to output. CAPs are used to define potentially causal relationships among input and output. In a feedforward neural network, the depth of the CAPs is equal to the network's distance, which is equal to the number of hidden layers plus one (as the

output layer is also parameterized). For recurrent neural networks, where a signal can pass through a layer multiple times, the CAP depth is theoretically infinite. [24]

There is no generally agreed-upon depth level that distinguishes shallow learning from deep learning, although most researchers accept that deep learning needs CAP depth greater than 2. In the sense that it can emulate any function, CAP of depth 2 has been shown to be a universal approximator. More layers do not improve the network's feature approximator capability. Extra layers aid in learning the features efficiently because deep models ( $CAP > 2$ ) can retrieve better features than shallow models. [25]

Deep learning architectures can be built using a greedy layer-by-layer approach [26]. Deep learning assists in disentangling these abstractions and determining which features increase results.

Deep learning approaches reduce function engineering in supervised learning activities by converting data into compact intermediate representations similar to principal components and generating layered structures that eliminate redundancy in representation.

Deep learning algorithms can be used to solve unsupervised learning problems. This is a significant advantage since unlabeled data is more plentiful than classified data. Neural history compressors and deep belief networks are two examples of deep systems that can be educated unsupervised. [27]

## 3. Convolutional Networks

Convolutional networks, also known as convolutional neural networks, or CNNs, are a type of neural network that is used to process data with a known grid-like topology. Time-series data, which can be thought of as a 1-D grid of samples taken at constant time intervals, and image data, which can be thought of as a 2-D grid of pixels, are two examples.

Convolutional networks have seen a huge amount of success in functional implementations. The term "convolutional neural network" refers to the network's use of a statistical operation known as convolution. Convolution is a subset of linear operations. Convolutional networks are essentially neural networks with at least one layer that uses convolution instead of general matrix multiplication. [11]

Isn't an image just a matrix of pixel values? Then why not simply flatten the image (for example, a 3x3 image matrix into a 9x1 vector) and pass it to a Multi-Level Perceptron for classification? This is not the case.

In the case of incredibly simple binary images, the system may display an average precision score when doing class prediction, but it would have little or no accuracy when dealing with complex images with pixel dependencies within.

Through the use of appropriate filters, a ConvNet can successfully capture the Spatial and Temporal dependencies in an image. Because of the reduced number of parameters involved and the reusability of weights, the architecture performs better fitting to the image dataset. In other terms, the network can be programmed to properly understand the image's complexity.

### 3.1. Convolution Layer

A convolutional layer is made up of a number of filters whose parameters must be known. The filters' height and weight are less than those of the input volume. Each filter is convolved with the input volume to generate a neuron activation map. To put it another way, the filter is slid over the width and height of the input, and at each spatial location, the dot products between the input and filter are computed.



The convolutional layer's output volume is calculated by stacking the activation maps of all filters along the depth axis. Since each filter's width and height are built to be smaller than the input volume, each neuron in the activation map is only connected to a small local area of the input volume. Each neuron's receptive field size is small and equal to the filter size. The architecture of the animal visual cortex, where the receptive fields of the cells are small, motivates the local connectivity.[12]

The convolutional layer's local connectivity enables the network to learn filters that adapt optimally to a local area of the input, thus leveraging the spatial local correlation of the input which is a pixel in an input image is more associated to adjacent pixels than to distant pixels. Furthermore, since the activation map is generated by convolution of the filter and the data, the filter parameters are shared over all local locations. Weight sharing limits the number of parameters for speech performance, learning efficiency, and strong generalization. [13]

In the Convolutional layer there are common functionalities Strides, Padding and Non-Linearity.

### **3.1.1. Strides**

The number of pixels that move around the input matrix is referred to as the Stride. As stride is set to one, the filters are shifted one pixel at a time. When the stride is 2, the filters are moved two pixels at a time, and so on.

### **3.1.2. Padding**

Padding is used where a filter does not exactly match the input image. We have two choices:

- To make the image suit, padding it with zeros (zero-padding).
- Remove the portion of the image that did not match the filter. This is known as true padding because it holds just the image's valid bits.

### 3.1.3. Non-Linearity

A sigmoidal nonlinearity function is often used in neural networks. However, there is growing evidence that other forms of nonlinearities can increase CNN efficiency. Rectified Linear Unit (ReLU) is an abbreviation for a non-linear operation. The output is  $f(x) = \max(0, x)$  [14].

#### 3.1.3.1. Advantages of the Rectified Linear Activation

When designing most forms of neural networks, the rectified linear activation function has quickly become the default activation function.

- **Computational Simplicity**

The rectifier function is simple to enforce since it only requires a  $\max()$  function. This is in contrast to the tanh and sigmoid activation functions, which involve an exponential measure.

*“Computations are also cheaper: there is no need for computing the exponential function in activations”* [17].

- **Representational Sparsity**

The rectifier function has the advantage of being able to output a real zero value. This is in contrast to the tanh and sigmoid activation functions, which learn to estimate a zero production, i.e., a value close to zero but not a true zero value.

This means that negative inputs will provide true zero values, allowing hidden layers in neural networks to be enabled with one or more true zero values. This is known as a sparse representation, and it is a valuable property of representational modeling because it can speed up learning while still simplifying the model.

Autoencoders, in which a network learns a compact representation of an input, such as an image or sequence before reconstructing it from the compact

representation, are one area that studies and seeks effective representations such as sparsity.

*“One way to achieve actual zeros in  $h$  for sparse (and denoising) autoencoders [...] The idea is to use rectified linear units to produce the code layer. With a prior that actually pushes the representations to zero (like the absolute value penalty), one can thus indirectly control the average number of zeros in the representation.”*  
[15]

- **Linear Behavior**

The rectifier function resembles and behaves primarily as a linear induction function. In general, a neural network's action is easier to optimize when it is linear or similar to linear.

*“Rectified linear units [...] are based on the principle that models are easier to optimize if their behavior is closer to linear.”* [16]

The trick to this property is that networks trained with this activation function almost entirely escape the issue of vanishing gradients since the gradients remain proportional to node activations.

*“Because of this linearity, gradients flow well on the active paths of neurons (there is no gradient vanishing effect due to activation non-linearities of sigmoid or tanh units).”* [17]

- **Train Deep Networks**

Importantly, the rediscovery and acceptance of the rectified linear activation mechanism made it possible to take advantage of hardware advancements and effectively train deep multi-layered networks with a nonlinear activation function using backpropagation.

As a result, bulky networks such as Boltzmann computers, as well as cumbersome training systems such as layer-wise training and unlabeled pre-training, could be left behind.

*“deep rectifier networks can reach their best performance without requiring any unsupervised pre-training on purely supervised tasks with large, labeled datasets. Hence, these results can be seen as a new milestone in the attempts at understanding the difficulty in training deep but purely supervised neural networks and closing the performance gap between neural networks learnt with and without unsupervised pre-training.” [17]*

It is also worth noting that there are several more non-linear activation layers, such as tanh or sigmoid, that can be used instead of ReLU. ReLU is used by the majority of data scientists because it outperforms the other two.

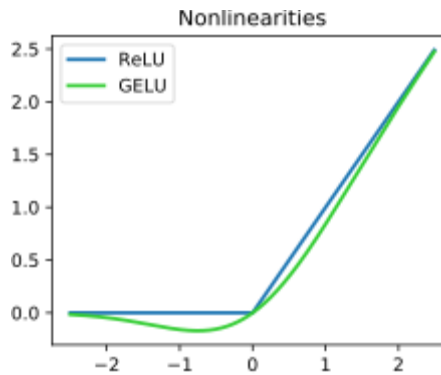


Figure 2. ReLU Activation Function vs GELU [14]

## 3.2. Pooling Layer

Convolutional layers in a convolutional neural network apply trained filters to input images in a systematic manner to construct feature maps that summarize the existence of certain features in the input.

Convolutional layers have shown to be very powerful and stacking convolutional layers in deep models helps layers close to the input to learn low-level features (e.g., lines) and layers deeper in the model to learn higher-order or more complex features, such as shapes or unique items.

One downside of convolutional layer feature map performance is that it records the exact location of features in the input. This implies that even minor changes in the location of the element in the input image result in a different feature map. This can occur when the input image is re-cropped, rotated, shifted, or other small adjustments are made.

Downsampling is a popular approach used in signal processing to solve this issue. This is the process of creating a reduced resolution image of an input signal that still includes the significant or essential structural elements but lacks the fine detail that might not be as helpful to the job.

Convolutional layers may be used for downsampling by varying the stride of the convolution across the picture. A pooling layer is a more stable and widely used solution.

A pooling layer is an additional layer that is applied after the convolutional layer. Specifically, after applying a nonlinearity (ex, ReLU) to the function maps output by a convolutional layer; for example, the layers in a model may look like this:

*(Input Image followed by Convolutional Layer then apply a Nonlinearity function and finally pass into Pooling Layer).*

A typical pattern for ordering layers within a convolutional neural network that may be replicated one or more times in a given model is to add a pooling layer after the convolutional layer.

The pooling layer works on each feature map independently to construct a new collection of clustered feature maps with the same number of features. Pooling is similar to applying a filter to function maps in that it entails choosing a pooling operation. The pooling

procedure or filter has a smaller scale than the function map; in particular, it is almost always 22 pixels added with a stride of 2 pixels. The pooling layer shrinks each function map by a factor of two, halving each dimension and lowering the number of pixels or values in each feature map to one-quarter of its original size. When a pooling layer is added to a feature map with a resolution of 6x6 (36 pixels), the result is a pooled feature map with a resolution of 3x3(9 pixels) [11].

Rather than being learned, the pooling operation is defined. The following are two traditional pooling functions, **Average Pooling** and **Max Pooling**.

The effect of employing a pooling layer and producing downsampled or pooled feature maps is a condensed representation of the features identified in the input. They are useful since minor variations in the position of the feature observed by the convolutional layer in the input result in a pooled feature map with the feature in the same location. The model's invariance to local translation is the functionality added by pooling.

*“In all cases, pooling helps to make the representation become approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change.”* [18]

A pooling layer is usually located between two convolutional layers. By downsampling the representation, the pooling layer reduces the number of parameters and computation. The pooling feature can be set to **maximum**, **minimum**, or **average**. Max pooling is widely used because it is more efficient. [13].

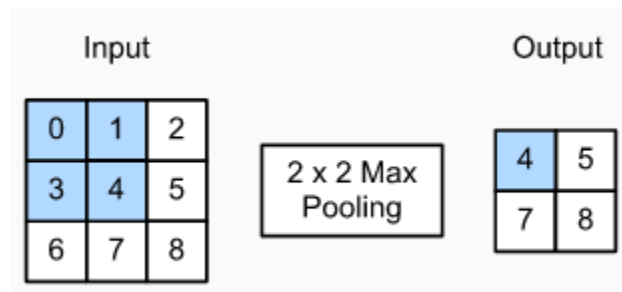


Figure 3. Max Pooling of 2x2 [28]

### 3.3. Fully Connected Layer

A completely connected layer is used at the output end of a standard deep neural network (DNN), such as a convolutional neural network. The convolutional neural network can be divided into two stages, which are **features extraction** and **classification**.

The feature extraction stage contains several different trainable completely convolutional (Conv) layers that learn high-level features, while the classifier portion, which is the fully linked layer, deals with various combinations of those features to make final decisions. Each completely linked layer, on the other hand, can be transformed into a convolutional layer. This is because the completely linked layer can be thought of as a sliding classifier that moves around the (Conv) attribute diagram. It is just that in most situations, the contribution is 1x1 spatial scale. As a result, any convolutional neural network is totally convolutional. [19]

Given a picture of size  $m \times n$  as input, the convolutional neural network with a completely connected layer at the end can be seen as producing a 1x1 sized output. What would happen if we feed it a  $k_m \times k_n$  picture instead? We can transform the final fully connected layers to convolutional layers by slipping the fully connected network over the final Conv layer, yielding a  $k \times k$  sized output instead. As a result, we can work with variable-sized input images by casting every completely connected layer as just another Conv layer. [19]

## 4. Transfer Learning

### 4.1. Overview

*“Transfer learning and domain adaptation refer to the situation where what has been learned in one setting ... is exploited to improve generalization in another setting” [35].*

As we see from the quoted definition above Transfer learning is a great machine learning technique that allows us to pass a model that was built for one task to be reused for other goals from a similar domain.

Moreover, it can be defined as an optimization process that allows rapid progress or improved performance when modeling different tasks.

*“Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned.” [36].*

There is a key feature that needs to be understood about transfer learning which is that transfer learning is not restricted to the deep learning field but what gives deep learning its popularity is the huge and needed resources to train deep learning models or in some cases the huge demanding datasets on which those models are trained.

### 4.2. Approaches to Transfer Learning [40]

There are three main uses for transfer learning:

- **Build a model to be reused**

Consider the situation in which you want to solve some “Task A” but lack the necessary data to train a deep neural network. Finding a similar “Task B” with a lot of data is one way to get around this. Using the deep neural network to train on task B and then use



the model to solve task A. The issue that is being tried to solve will determine whether it needs to use the whole model or just a few layers.

If the feedback in both tasks is the same, you might reuse the model and make predictions about the new input. Changing and retraining various task-specific layers and the performance layer needs to be investigated.

- **Using a pre-trained model**

The second solution is to use a model that has already been trained. There are a couple of pre-trained models that have been published, however, to use a pre-trained model some analysis needs to be done first, Like the number of layers to reuse and retrain is determined by the given problem.

- **Build a model to extract features**

Another approach is to use deep learning to find the best representation of a given task, which entails finding the most important elements. This method, also known as representation learning, will often produce much better results than designing representations from scratch.

Typically, features of machine learning are hand-crafted by developers and domain experts. Deep learning, fortunately, will retrieve functionality automatically. Of course, this does not negate the importance of feature engineering and domain awareness there is always a need to determine which functionality to include in the network. Having said that, neural networks will learn which features are reassuring.

Neural networks, on the other hand, have the potential to learn which functions are critical and which are not. Also, for complicated tasks that would otherwise necessitate a lot of human work, a classification learning algorithm will discover a successful combination of features in a very small time frame.

The studied representation will then be applied to other problems. Simply use the first layers to identify the correct function representation, but do not use the network performance because it is too task-specific. Feed data into the network instead, by

using one of the hidden layers as the output layer. This layer could then be represented as a raw data representation.[37]

This method is often used in computer vision since it reduces the size of the dataset, which reduces processing time and makes it more efficient for conventional algorithms.[37]

## 5. Related Work

### 5.1. Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning [9]

Using a high pixel-level supervision approach and a relatively limited training dataset, they were able to demonstrate an end-to-end network that performs mutual classification and segmentation in this work. that, according to them, has the best classification accuracy to date [9] as compared to other deep learning techniques, and simultaneously localizes these anomalies. They showed that it detects several anomalies that experts overlook. Furthermore, they showed positive results for multiclass hemorrhage segmentation while retaining correct diagnosis at the inspection stage.

They used a single-stage network for combined segmentation and examination-level classification to solve the need for both correct examination-level classification and parallel localization of anomalies, which has the following advantages:

- At both preparation and testing times, only one network is used for segmentation and inspection grouping, rather than two.
- There is a high degree of feature exchange between segmentation and classification networks. In general, sharing the representation between correlated tasks is advantageous since it saves computation and serves as an efficient regularization process [9]. The construction of the hemorrhage detection mechanism is depicted in Figure 4.

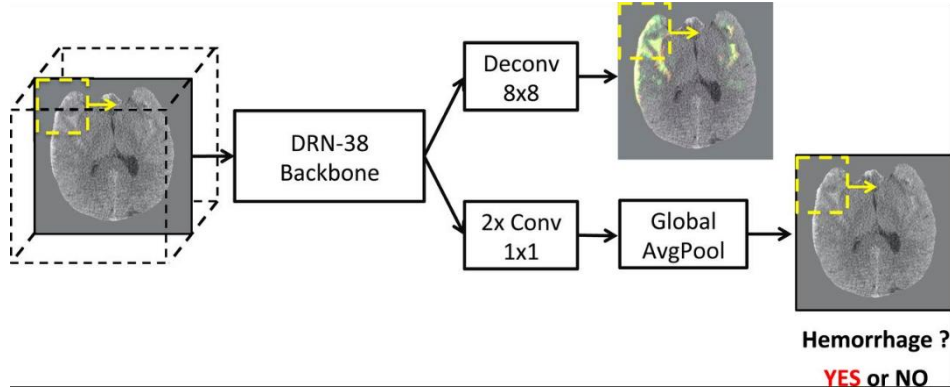


Figure 4. hemorrhage detection system architecture [9]

In summary, they demonstrated a deep learning algorithm built on a strong supervision methodology and a relatively limited training dataset for detecting and localizing acute intracranial hemorrhage on head CT. They demonstrated results equal to highly qualified professionals. PatchFCN can be useful for deriving quantitative biomarkers from CT and other radiological tests, in addition to the main clinical role of classifying head CT examinations as positive or negative for anomalies.

## 5.1.1. Methods and Materials

### 5.1.1.1. Data preprocessing

The skull and face were separated from CT images using a combination of image processing methods, including thresholding, to distinguish skull and facial bones, supplemented by a series of close, open, and fill operations to preserve only intracranial structures. Individuals may, in principle, be detected by the surface rendering of facial soft tissue pixels found in the original data, which improved data protection. It also simplifies the issue for the network since it just has to model intracranial structures.

### 5.1.1.2. Network architecture

PatchFCN, a completely convolutional neural network (FCN), was the aim of the architecture (9). It is an FCN with modifications chosen after a period of algorithm creation in which many architectures were explored:

- In order to mimic radiologists' method of adjudicating tiny hemorrhages by using contextual information in slices adjacent to the image of interest, it uses three feedback channels composed of “flanking” images immediately superior and inferior to the image under evaluation.
- 2. The network evaluation on any single pass is restricted to a subset or "patch" of the image to model x–y axes background, forcing the network to make decisions based on more informative local image knowledge.
- It contains a patch classification branch [6] to separate the patch prediction from the noisier pixel predictions and to improve patch prediction accuracy. Figure 4 depicts the whole structure.

### 5.1.1.3. Implementation details

Dilated ResNet 38 was used as the network backbone architecture, and all hyperparameters were built using the UCSF-4.4K training package. They used stochastic gradient descent (SGD) and a momentum of 0.99 to optimize cross-entropy loss. Every 160 epochs, the learning rate was reduced by 0.1.

To monitor the class imbalance between positive and negative cases in the testing dataset, they sampled 30% of the patches from positive images in each training minibatch and increased the weighting of positive pixel loss by a factor of 3. Since the multiclass tests were exploratory in nature, no balance across positive class categories was done.

The backbone and the pixel estimation branch (1 up-convolution layer) were trained for 400 epochs at an initial learning rate of  $1e-3$ . The patch classification branch (Conv + Batchnorm + ReLu + Conv layers) was then conditioned for 40 epochs. Finally, the entire model was fine-tuned collaboratively for 30 epochs at a learning rate of  $5 \times 1e-5$ . At inference time, neighboring patches were sampled such that they overlapped by two-thirds.

All patch's pixel predictions were converted to image space and summed to produce the final prediction. The highest patch classification score in the stack was taken as the stack classification score. On average, the model tests each stack in 1 second. [9]

#### **5.1.1.4. Multiclass architecture**

They performed an exploratory analysis at the pixel and examination stages on the multiclass estimation of hemorrhage forms. The model performance layers were revamped for the following tasks:

- Instead of two output channels, the pixel classifier has  $N + 1$ , where  $N$  is the number of hemorrhage classes.
- For the  $N$  hemorrhage groups and the cumulative positive class, the stack classification branch produces  $2(N + 1)$  outputs. The discovery that the classes are mutually exclusive at the pixel level (i.e., each pixel is a part of only one class, or subtype, of hemorrhage) but not at the inspection level inspired this design (i.e., each examination can contain multiple classes of hemorrhage).[9]

#### **5.1.1.5. Dataset**

They used a training sample of 4,396 head CT scans to build the algorithm. This dataset (UCSF-4.4K) contains 1,131 positive intracranial hemorrhage tests and 3,265 negative examinations. The training dataset included a wide range of hemorrhage sizes and forms, as well as imaging artifacts, and was obtained from four separate CT scanners from two main CT vendors (GE Healthcare and Siemens Healthiness) between 2010 and 2017. Each inspection included a 3D stack of 27 to 38 transverse 2D images acquired via the head on 64-detector row CT scanners. Two ABR-certified radiologists with a CAQ in neuroradiology independently checked the pixelwise labels for acute intracranial hemorrhage [6].

They obtained a different research sample of 200 head CT scans to verify the algorithm. The distribution of positive and negative cases across system forms is seen in Figure 5 for both the training and test sets.

They aimed for an average positive rate of 10% to 15% for acute intracranial hemorrhage, which is comparable to the positive head CT rate, when developing the test collection.

Table 2.  
Training and test datasets

Dataset	Result	GE	Siemens	Neurologica	Total
Training	Positive	821	310	0	1,131
	Negative	768	2,497	0	3,265
Test	Positive	0	25	0	25
	Negative	14	160	1	175

Figure 5. Distribution of positive and negative cases across the Dataset [6]

### 5.1.2. Evaluation of model performance

To assess model results, the deep learning algorithm was run exactly once on the test set of 200 CT examinations, with no changes made to the hyperparameters chosen during the algorithm development process. This eliminated the risk of overfitting to the test results because the recorded output could closely reflect the model's true performance. The algorithm shows pixel-level and examination-level odds (continuous from 0 to 1) for the presence of intracranial hemorrhage for each scan in the research dataset of 200 CT exams. Including the fact that certain patients have two or three head CT scans at the same hospitalization, it was made certain that each patient appeared only once in either the preparation or test sets, but not both.[9]

The deep learning algorithm's receiver operating characteristic (ROC) was measured and compared to the "gold standard" to recognize the occurrence of acute intracranial hemorrhage on each CT inspection. A cautious consensus interpretation by two ABR-certified neuroradiologists with a CAQ in neuroradiology was the gold standard for interpreting all 200 CT scans in the research collection as positive or negative for acute intracranial hemorrhage.

## **5.2. Precise diagnosis of intracranial hemorrhage and subtypes using a three-dimensional joint convolutional and recurrent neural network [34]**

This study developed a system that is an end-to-end trainable network with the ability to train under two different levels of annotation details: (1) Only ground truths for topics (i.e., marks for the whole scans) are available, and (2) ground truths for any of the scans' slices are available.

### **5.2.1. Methods and Materials**

#### **5.2.1.1. Image Preprocessing**

A series of preprocessing operations were carried out prior to feeding the data to the model to be trained. To minimize GPU memory consumption, all images were resampled to 512x512 pixels and then downsampled to 256x256 pixels. Knowing this, each initial scan was assigned a number. Three separate contrast windows were used to normalize the images in order to accommodate for the high dynamic intensity spectrum while preserving the information for different objects of interest.

#### **5.2.1.2. Dataset**

Head CT scans from 3129 participants were originally obtained from three independent hospitals, with the knowledge that both subjects were Asian.

Following a meticulous slice-by-slice examination and annotation by three independent professional radiologists (with 10-, 12-, and 16-years of experience in analyzing head CT scans, respectively), 293 cases were omitted from further study due to insufficient detail or significant imaging objects.

In the end, we used the remaining 2836 cases in our report, which included 1836 subjects with ICH and 1000 healthy subjects. They maintained such a high ICH prevalence in 65 percent of the dataset to ensure there were enough positive samples to support the algorithms' learning process.



## 5.2.2. Training and evaluating models

### 5.2.2.1. Prediction models

The model was designed to detect both two-type classification and five-type classification. To be more precise, the algorithm is built on an architecture that includes a CNN component followed by an RNN component to simulate how radiologists interpret scans. The CNN component extracts valuable features from the slice, while the RNN component uses those extracted features to calculate the likelihood of ICH or a subtype.

To improve classification accuracy, the RNA architecture is used to collect sequential knowledge of features from consecutive slices by adding interslice dependency background. The model workflow for detecting ICH subtypes is depicted below.

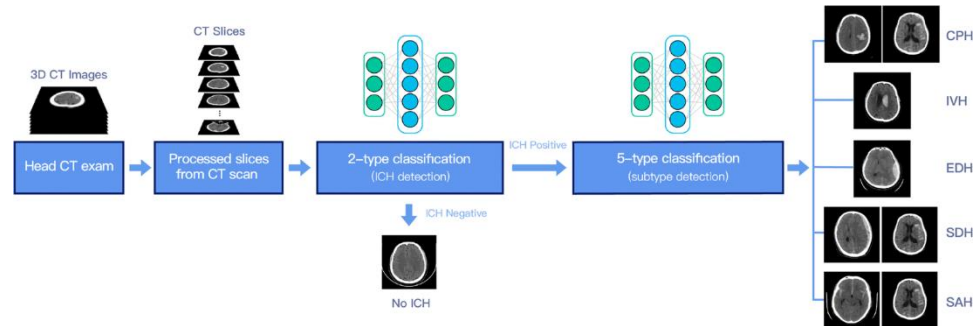


Figure 6. illustration of the model workflow of detecting ICH subtypes.[34]

The model was designed to perform two-type classification to decide if ICH was present in a slice. If the slice was predicted to be ICH-positive, a multi mark classification step was conducted to determine which of the five ICH subtypes it belonged to. The process can be visualized in Figure 6 above.

#### **5.2.2.2. Training procedures**

The dataset was randomly divided into three parts: training (80%), validation (10%), and testing (10%). The training set was used to refine the model parameters, the validation set was used to avoid overfitting, and the testing set was used to evaluate the model.

#### **5.2.2.3. Results**

For Two type classification "Binary classification" achieved 98 percent accuracy with minor variations in all evaluation criteria, and "multi-label classification" achieved 98 percent accuracy with five types of classification. Of the five subtypes, CPH had the highest proportion of positive cases, while EDH had the lowest (CPH > SAH > IVH > SDH > EDH). Across both subtypes, the algorithm had an AUC greater than 0.8 and a precision greater than 0.8.

During the training, significant findings were made depending on the metric's sensitivity. CPH was by far the most performed subtype, with sensitivity values greater than 0.9. Furthermore, also with the model conditioned with slice-level marks, the sensitivity for SAH and EDH was just 0.69, clearly lower than the other three subtypes.

The low sensitivity score of SAH may be attributed to the difficulties of diagnosis, since it is considered the most difficult subtype to diagnose, while the low sensitivity score of EDH may be due to the incredibly low number of positive cases: only 6.4 percent (94/1461) of the subjects and 1.9 percent (758/39,278) of the slices are EDH-positive.

### 5.3. Convolutional neural networks for detection intracranial hemorrhage in CT images [29]

This study presents the use of convolutional neural networks for the purpose of classifying hemorrhage vs non-hemorrhage. 491 trials of computed tomography of the head without comparison were used for a total of 193,317 slices of which there are four forms of intracranial hemorrhage and brains stable were used.

#### 5.3.1. Methods and Materials

##### 5.3.1.1. Image Preprocessing

For the original non-contrast CT series dataset, they first agreed to delete the background image from all slices because it provides little detail to the classification algorithm. Instead of using the whole CT dynamic spectrum, the densities of each slice is windowed using a window (level=50, width=80) to visualize only the brain parenchyma.

An anisotropic filter with values (kernel=0.02, time=10) was used. Additionally, the pixel values in each slice of the dataset were standardized between 0 and 1. Finally, the image is resized to 256 x 256 pixels. Before transferring the data to the deep learning models, all preprocessing was completed. Figure 6 depicts CT slices with and without hemorrhage after the preprocessing point. [29]

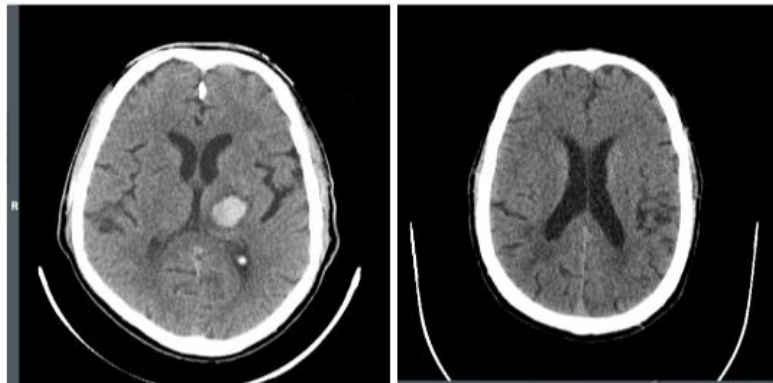


Figure 7. CT slices with the presence of hemorrhage and no presence after preprocessing stage [29]

### 5.3.1.2. Dataset

The dataset selected is known as CQ500, and a total of 491 studies were analyzed by three independent specialist radiologists with 8, 12, and 20 years of experience in cranial CT image analysis.

The Dataset has the following characteristics: [29]

Characteristic	CQ500 dataset
No. of scans	491/193.317slices
Mean age.	22.43
No. of scans (percentage) with intracranial hemorrhage	205(41.17%)
Intracerebral	162(32.99%)
Subdural	53(10.79%)
Extradural (Epidural)	13(2.64%)
Subarachnoid	60(12.21%)

## 5.3.2. Convolutional Neural Network

In this study, two convolutional neural networks were used for hemorrhage detection: first, a basic custom CNN was developed while retaining parsimony. Figure 7 depicts the network architecture of the first CNN, while Figure 8 depicts a modified version of the popular VGG 16 model.

### 5.3.2.1. Custom CNN Model

- Two convolutional layers with ReLu activation and kernel (3x3)
- Max-pooling layer size reduction
- Two convolutional layers with the same characteristics than previous convolutional layer

- Followed by a max-pooling layer with kernel(2x2)
- Flatten layer to prepare the features maps to dense layers.
- Finally, two fully connected layers for classification was implemented to predict labels with sigmoid activation.

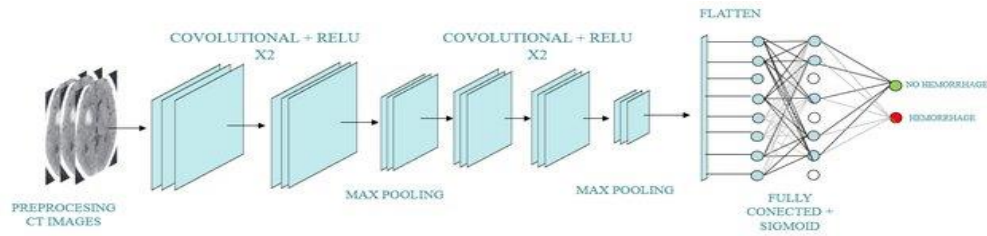


Figure 8. CNN Model architecture for binary classification [29]

#### 5.3.2.2. Modified VGG 16

- Modification for binary classification (hemorrhage vs no-hemorrhage)
- 5 blocks (convolutional layer + pooling layer)
- 3 fully connected layers used for the classification task.

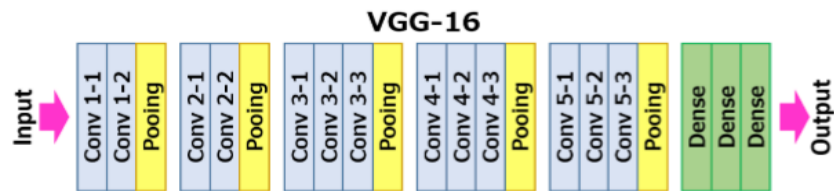


Figure 9. VGG model architecture [29]

### 5.3.3. Training and evaluating models

The CNN models were trained using preprocessing slices from 491 CT scans, with a total of 193,317 slices used to train and test the CNN network. We then suggested two strategies for training the models:

- Slices randomized:

Regardless of subject independence, all slices were randomized to train (0.85) and evaluate (0.15) sets. A subject's slices could be in the train package, while another portion of the slices could be in the test set.

- Subject randomized:

All slices were randomized to train (0.85) and test (0.15) sets, maintaining subject independence; this ensures that all slices from the same subject were sent to train or test.

Divide the dataset by 0.85 for preparation, with the remaining 0.2 used for evaluation during the training phase. In this case Each model was trained for 150 epochs in batches of 32, and the best model was saved to be tested with the test set. Accuracy, recall, and F1 measures, as well as ROC curves, were obtained for each of the “CNN4”, “VGG 16” algorithms using binary cross-entropy loss.

## 6. Our solution

### 6.1. Dataset

Data from the “RSNA Intracranial Hemorrhage Detection <sup>[41]</sup>, Identify Acute Intracranial Hemorrhage and its Subtypes” task was used. It includes 752 803 DICOM files. Each file represents a slice of the pie. And the files are 512 x 512 in size. A CSV file containing metadata for the corresponding DICOM files is provided for the results.

CSV has 4516842 rows and 2 columns, as seen in Figure 9, with the ID column containing the ID of the slice and the subtype of ICH appended to it, and the Label column containing a binary value indicating if the related ID in the same row has ICH.

	ID	Label
0	ID_12cad6af_epidural	0
1	ID_12cad6af_intraparenchymal	0
2	ID_12cad6af_intraventricular	0
3	ID_12cad6af_subarachnoid	0
4	ID_12cad6af_subdural	0
5	ID_12cad6af_any	0
6	ID_38fd7baa0_epidural	0
7	ID_38fd7baa0_intraparenchymal	0
8	ID_38fd7baa0_intraventricular	0
9	ID_38fd7baa0_subarachnoid	0

Figure 10. CSV File annotation

Upon updating it to make it more useful, It is made up of 674257 rows and 6 columns, with each column representing one of the ICH Subtypes. We can see from the number of rows on the updated CSV in Figure 10 that there are 78 546 DICOM files without annotation.

	any	epidural	intraparenchymal	intraventricular	subarachnoid	subdural
filename						
ID_000039fa0	0	0	0	0	0	0
ID_00005679d	0	0	0	0	0	0
ID_00008ce3c	0	0	0	0	0	0
ID_0000950d7	0	0	0	0	0	0
ID_0000aee4b	0	0	0	0	0	0
...	...	...	...	...	...	...
ID_ffff73ede	0	0	0	0	0	0
ID_ffff80705	0	0	0	0	0	0
ID_ffff82e46	0	0	0	0	0	0
ID_ffff922b9	1	0	0	1	0	0
ID_ffff9393	0	0	0	0	0	0

Figure 11. CSV Table of columns for separating the ID and the ICH Subtypes

After applying general EDA (Exploratory data analysis) on the Data we noticed that the Dataset is imbalanced as shown in Figure 11 and Figure 12.

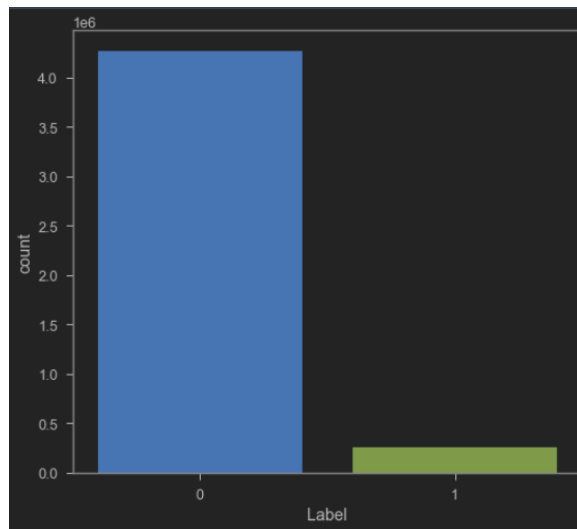


Figure 12. General lookup at the percentage of Hemorrhage vs No- Hemorrhage Slices



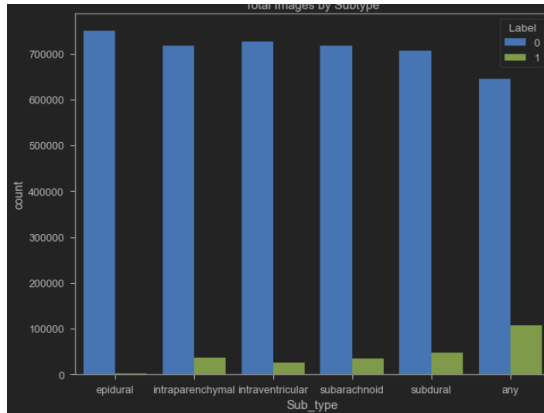


Figure 13. General lookup at the percentage of Hemorrhage vs No- Hemorrhage Slices with their Subtypes

While trying to analyze the DICOM files we found that the one-pixel array of a file is corrupted “ID\_6431af929.dcm” and 78 546 out of 752 803 DICOM file without annotation “Out of use”.

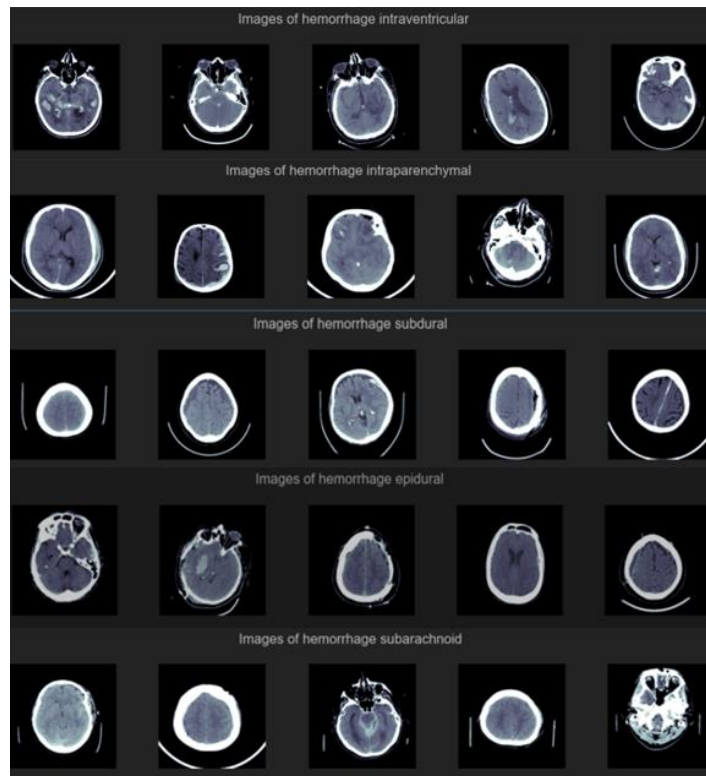


Figure 14. Some examples of ICH cases in our Dataset

## 6.2. First experiment

### 6.2.1. Image Preprocessing

To achieve optimal training performance, we used image preprocessing techniques to eliminate the background image from all slices and remove unwanted noise.

Pixel Normalization was used in conjunction with Pixel Centering to scale pixel values to have a zero mean and unit variance, and Pixel Standardization to scale pixel values to have a zero mean and unit variance.

### 6.2.2. Model Architecture

We build a model for binary classification to detect if the brain image has Hemorrhage or it is a healthy brain.

The model has two structures and can be seen in Figure 15:

- Feature Extraction part:  
it consists of two Convolutional 2d layers with kernel size 8 x 8 and stride 1 x 1 and each of them is fed to ReLU activation Function to add non-linearity the First Conv layer has an input of 1 color channel and 16 input size while the second Conv layer has an input of 16 and output of 32, following to the ReLU function there is a max-pooling layer with kernel size 2 x 2 and stride of 2 with no padding and dilation of 1, after that the output will be passed to a flatten Layer.
- Classification part:  
The Flatten layer will be fed to 3 Fully connected layer and each of these layers will be fed to the ReLU activation function and between these fully connected layers, there are Batch Normalization layers and right before feeding it to the output Fully connected layer, we added Dropout function to prevent the model from over fitting.

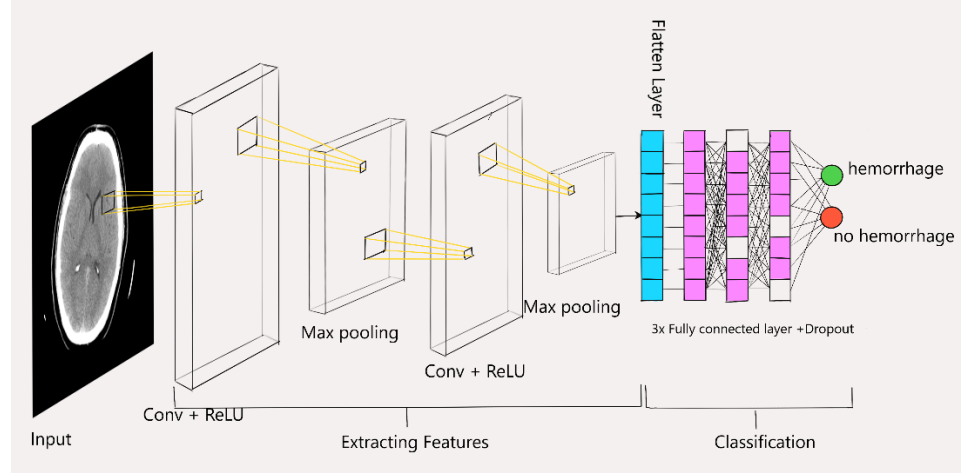


Figure 15. Our CNN model architecture for binary classification

### 6.2.3. Training and Validation

For training we split the dataset randomly which is 674,257 slices to 80% percent for training and 20% for testing, we trained the model for 50 epochs with batch size of 32 and learning rate of  $1e-3$ , for loss function we used BCEWithLogistLoss function for binary classification and we used Adam Optimizer, we got an accuracy of 63.58% on the test data and the following results in the table down below:

Accuracy	Recall	F1 measure	ROC
<b>0.636</b>	0.67	0.71	0.761

## 6.3. Second Experiment

### 6.3.1. Image preprocessing

With EDA completed, we proceeded to process our dataset for further modeling. Here, we were introduced to the concepts of windowing and Hounsfield Units. Tissue densities in CT images are measured in Hounsfield units (HU). HUs usually vary from -1000 to +1000, with -1000 representing air and +1000 representing bone. [31]

Although a computer display can detect 256 shades of grey in a CT picture, the human eye can detect only 6% of the greyscale, or approximately 17 shades of grey. The human eye detects a difference in about 117 HUs, resulting in substantial detail loss in medical diagnosis. [32]

We opted to use windowing to handle this exercise. Windowing is often used by surgeons, and we used it to optimize the knowledge submitted to the model. It enables one to concentrate on a limited number of HUs in the vicinity of the hemorrhage territory for more accurate identification. Since we discovered variations in the HU spectrum for various subtypes, we built three windows: brain, soft, and subdural, and fed them into our model's three RGB channels.

We were able to differentiate ICH subtypes that may have been missing in a CT scan image by creating three custom windows. Figure 15 reveals a CT scan of subdural hemorrhage with minor irregularities at the skull surface. This is visible in the subdural window (shown below), but not in the others.

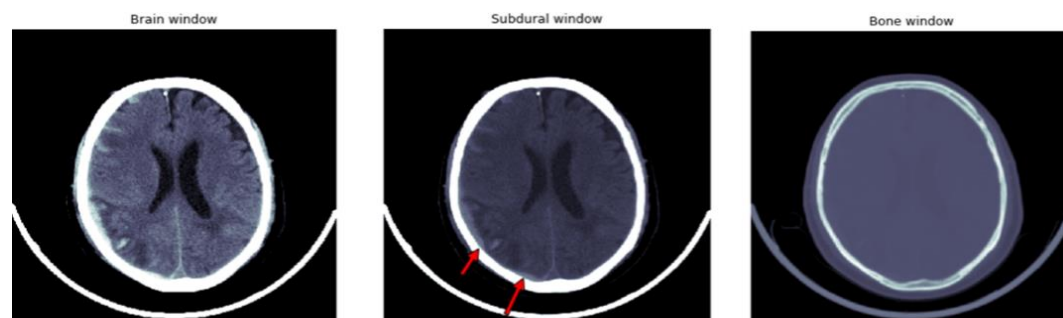


Figure 16. The effect of using different Windows to detect ICH Subtypes.

To put this idea into practice, we used the DICOM file's rescale slope and intercept parameters to transform the pixel data array into Hounsfield units. The glass center and width of the image are then used to define the start and endpoints of the window. The data from these windows are then stacked and returned as RGB channel output.

Before applying any other steps, we noticed that the data is not clean, DICOM slices were taken from different positions which led to inaccurate results during training, so we arranged the slices in ascending order according to the slice position Z which is labeled in the DICOM header as “Position2” and then group those slices by “StudyInstance” which also can be found in the DICOM file header, to make sure that we separate all the slices that were taken from the same Study Instance.

Furthermore, since we know that each slice was taken by 5mm long, we decided to separate some of the slices that are placed in the center of one study case. By doing so, we ensured that the images that will be fed to the model will not differ that much, resulting in more robust and smooth learning, and decreased fuzz in the dataset.

That being said, we isolated 5 slices above the middle slice and then produced a training set and a testing set at random from the total dataset in an 80/20 ratio.

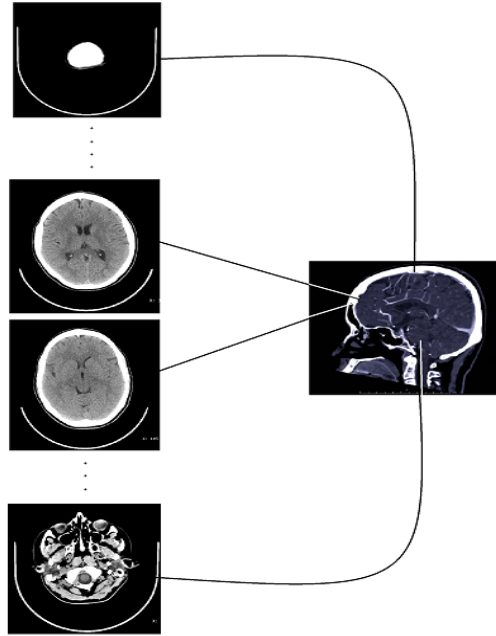


Figure 17. Illustration of choosing slices to create the dataset.

### 6.3.2. Balancing Data

We often come across datasets in many machine learning systems where certain kinds of data appear more often than others. Consider the discovery of rare diseases, there are likely more common samples than disease samples.

Moreover, a commonly used technique known as resampling is used. It entails eliminating samples from the majority class (under-sampling) and/or using more examples from the minority class (over-sampling). Despite the benefit of applying balancing techniques to classes, these strategies have several drawbacks.

The under-sampling methodology creates a new subclass from the initial dataset by excluding certain samples from the majority class.[38]

The over-sampling process generates a new superset from the original dataset by duplicating any of the minority class's samples or creating new samples based on the minority class's original samples.[38]

A hybrid approach is one that combines over-sampling and under-sampling techniques.[38]

To address this problem, we measured the weight of each class in the dataset and then build the weight class distribution, later when sampling from an imbalanced sample, apply rebalancing on the class distributions using a hybrid method between under-sampling and over-sampling which can be seen in the Figure 18.

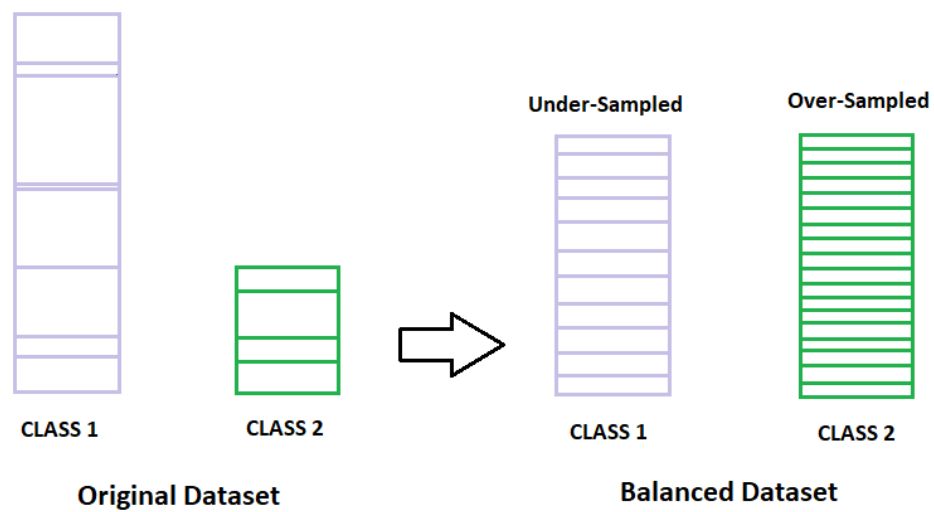


Figure 18. Illustration of data-based approaches, which includes a hybrid method of under-sampling, over-sampling.

### 6.3.3. Using Transfer Learning for Classification Task

We decided to use prebuilt models and make the requisite modifications to suit our problem after conducting a thorough review of our challenge. Since we can reap the advantages of transfer learning, we can reduce training time, improve neural network efficiency (in most cases), and avoid the need for a large amount of data to get the desired outcomes.

Here is how it functions in basic terms:

First, we start by removing the final layer used to create predictions, known as the "loss output" layer, and replace it with a new loss output layer for "Class" prediction. This output loss layer is a fine-tuning node for deciding how training penalizes deviations from labeled data and predicted output.

Then, we would take our ICH dataset and train it on all of the neural network's layers, or just the last few layers, or just the loss layer. The output of the latest CNN would be ICH classification as a result of the application of these transfer learning strategies.

To build our model we used a well know model called "ResNet 152" which has 152 layers that utilizes the residual blocks and has a central concept of introducing a so-called "identity shortcut connection" that bypasses one or more layers.[39]

We used the ResNet 152 model architecture to predict which images have ICH and which do not use "Binary classification" model, and we used the "Multi-label classification" model to classify which subtype of ICH is present in a given image.

We started by applying preprocessing techniques mentioned above in "6.3.1 Image preprocessing" and then resale the image size from 512x512x3 to 256x256x3 and applying random transformation to get the final size of 224x224x3, before directly feeding the modified dataset to the model we had to apply changes to the original ResNet-152 architecture to fit our model, that being said, For Binary classification step we kept the feature extraction layers and removed the last classification layer and replaced with our own, that has the following architecture in Figure 19:



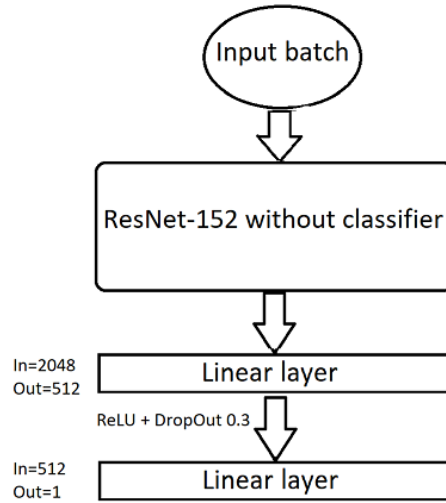


Figure 19. Illustration of Binary classification model

And for Multi-label classification task we followed similar technique to what we did in the Binary classification but instead of making one classification layer for all classes we built five parallel binary classification layers for each class “Subtypes of ICH” and pass to them the images according to the class label, and this can be demonstrated in Figure 20 below:

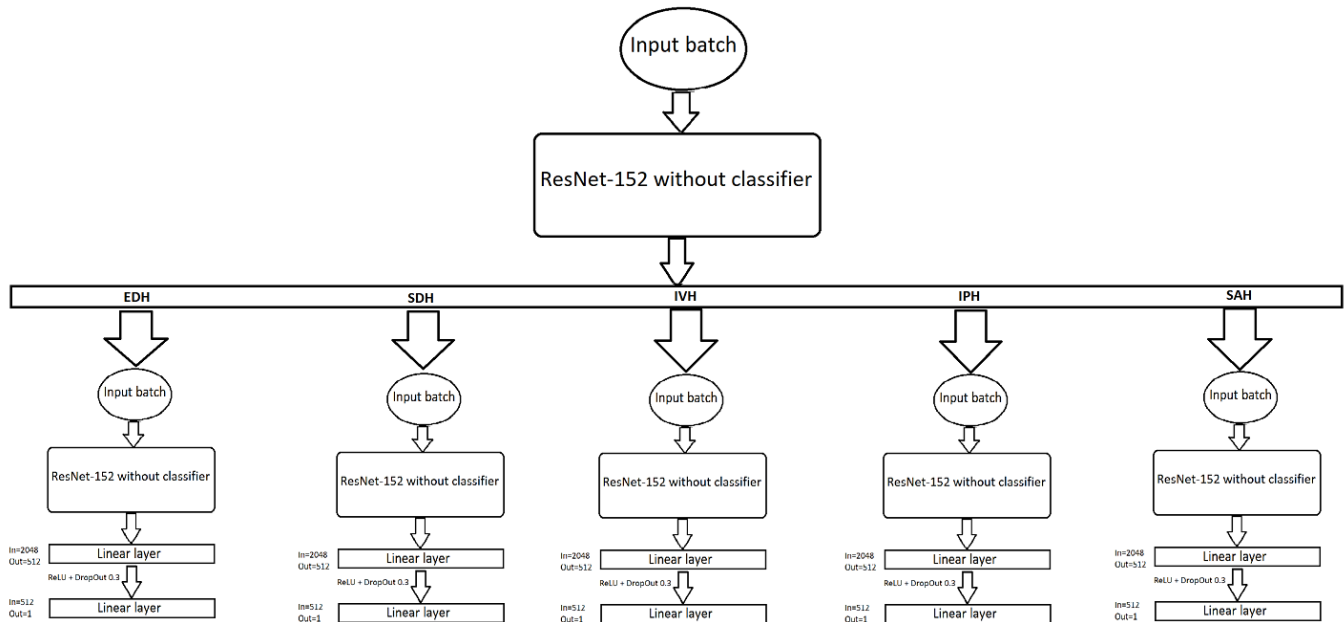


Figure 20. Illustration of Multi-label classification model

By building those models we ensured full coverage for both classification scenarios, to start training the models we started by applying preprocessing step and that generated for us a training set and a test set for both of “Binary classification” task and the “Multi-label classification” task, knowing that for multi-label Dataset, we isolated just the images that have ICH, Datasets can be seen in the following table below:

	<i>Train Dataset</i>	<i>Test Dataset</i>
<i>Binary classification Dataset</i>	78116	19529
<i>Multi-label Classification Dataset</i>	22501	5756

Furthermore, we built batches of the Binary classification datasets to train the Binary classification model whether the image has ICH or not and validate the learning outcomes at each epoch. To pass a batch to the model, it must first pass through a “Data-sampler class” (mentioned in 6.3.2 Balancing Data) to ensure that our model is not biased to one class more than the other because we have a lot more of the healthy images than hemorrhage ones.

Then, we continued on to the next stage, which is training the Multi-label classification model. Here, we took a different approach: we generated batches from the “Multi-label dataset” that only included hemorrhage pictures, which were then transferred to the Multi-label classification model, which taught the model to detect ICH subtypes in parallel and then validated those outputs in the test dataset at each epoch.

The previous procedure can pretty much be summed up in the illustration shown in Figure 21 below:

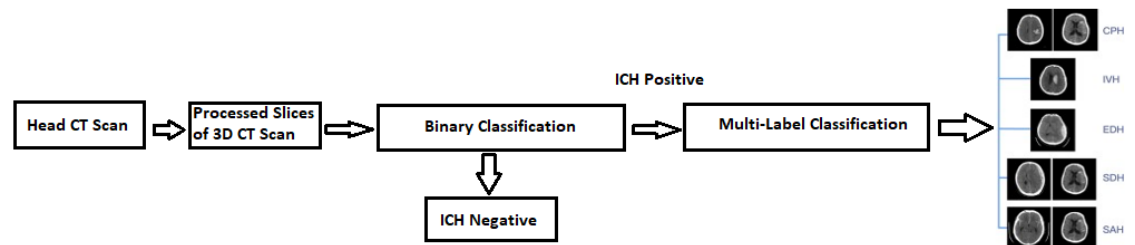


Figure 21. Illustration of the workflow of the Solution

## 6.3.4. Training and Validation Results

### 6.3.4.1. Binary Classification's Results

After training the model on 78116 images for training and 19529 for testing and applying the following Configuration for the Hyperparameters:

<i>Name</i>	<i>Train Batch Size</i>	<i>Test Batch Size</i>	<i>Epoch</i>	<i>Learning Rate</i>	<i>Momentum</i>	<i>Optimizer</i>
<i>Value</i>	32	32	50	1e-3	0.9	SGD

We got F1 accuracy score of 84% across the classes with minor variations in all evaluation criteria, and the results for accuracy and loss values can then visualized down below in the graphs:

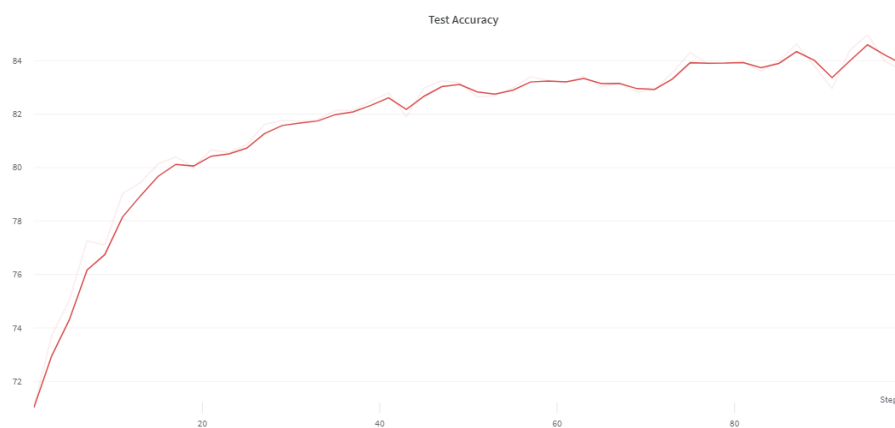


Figure 22. Test accuracy of F1 score on testing data after each epoch

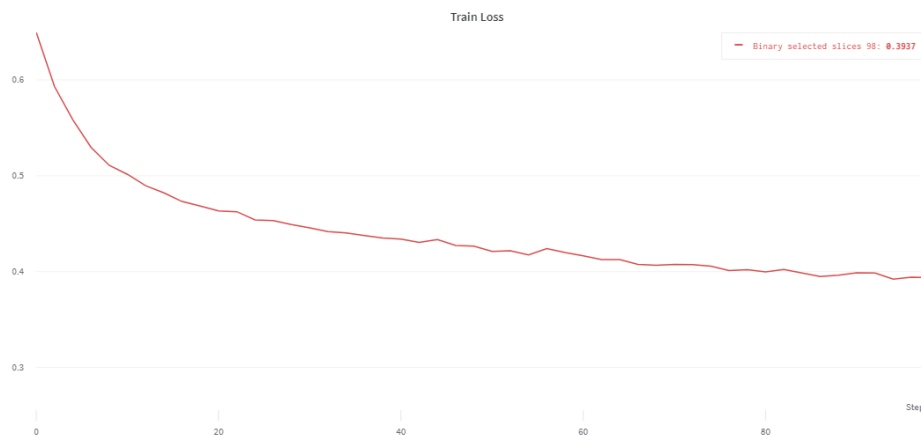


Figure 23. Train Loss score on testing data after each epoch

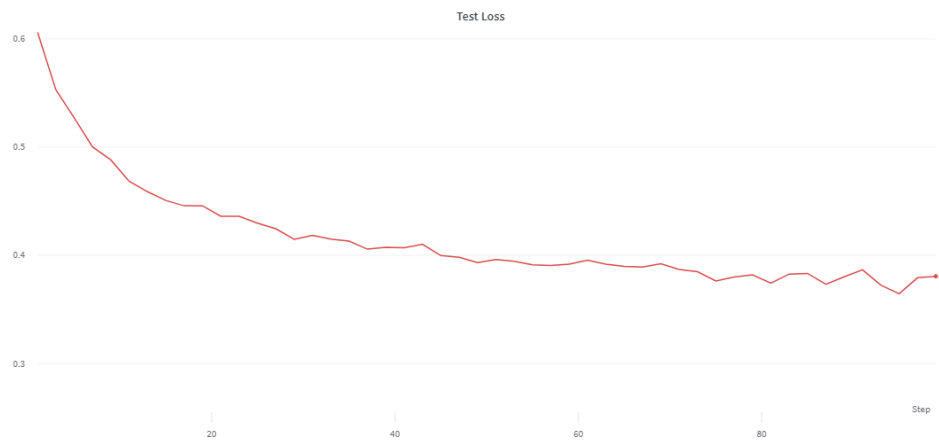


Figure 24. Test Loss score on testing data after each epoch

We can see some of the results that our model has predicted to be healthy or has ICH:

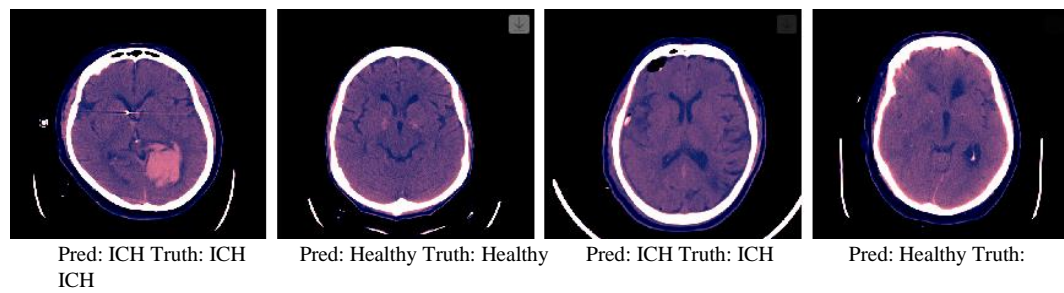


Figure 25. Some Predictions of Binary classification

### 6.3.4.2. Multi-Label Classification's Results

For this task we applied similar configuration with some changes to get better results, so we applied three different configurations on the Hyperparameters to reach optimized points:

<i>Name</i>	<i>Train Batch Size</i>	<i>Test Batch Size</i>	<i>Epoch</i>	<i>Learning Rate</i>	<i>Momentum</i>	<i>Optimizer</i>
<i>First Run</i>	32	32	150	1e-4	0.9	SGD
<i>Second Run</i>	32	32	100	1e-3	0.9	SGD
<i>Third Run</i>	32	32	50	1e-2	0.9	SGD

In this task we applied a Learning rate decay technique that allows dynamic learning rate reducing based on some validation measurements, which in our case was the testing loss results after each 2 epochs if there were not any improvements it will decay the learning rate.

By applying that we got an accuracy of 84.5% average across all classes, and the results can be seen below in the graphs for all three runs with different configurations.

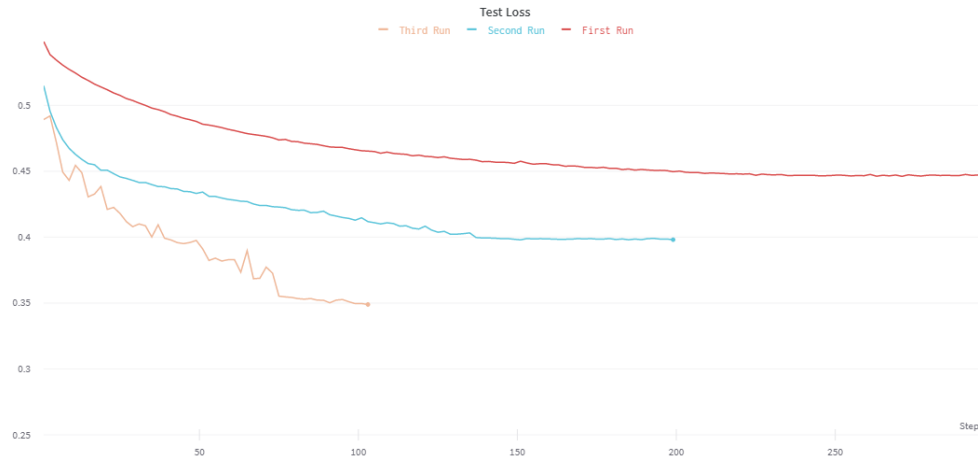


Figure 26. Test Loss score on testing data after each epoch

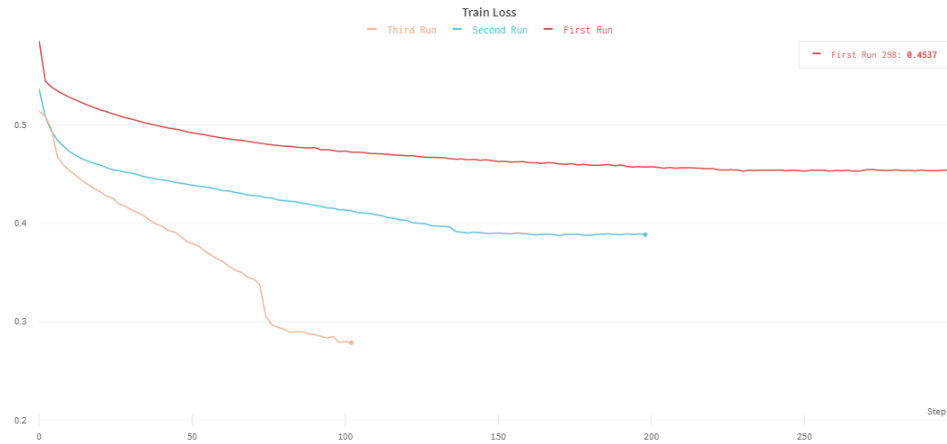


Figure 27. Train Loss score on testing data after each epoch

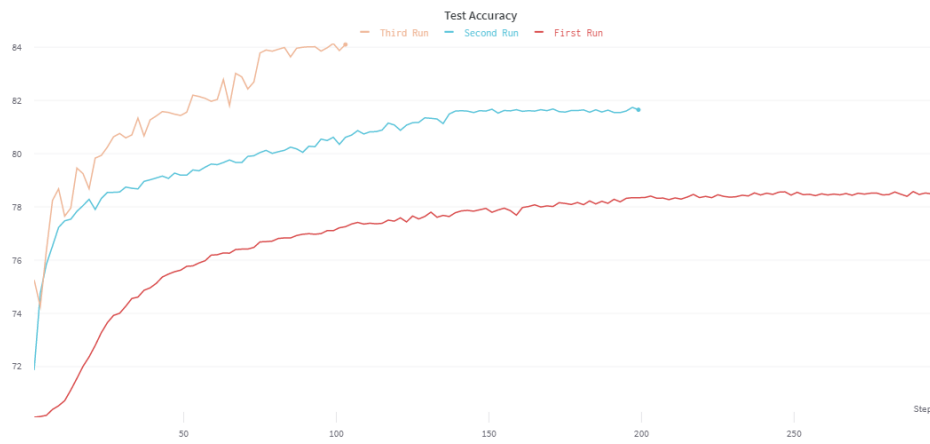


Figure 28. Test accuracy score on testing data after each epoch



Figure 29. Some Predictions of Multi-label classification

During the training, significant findings were made depending on the metric's sensitivity. Subdural ICH was by far the most performed subtype. Furthermore, also with the model conditioned with slice-level marks, the sensitivity for Epidural ICH, which was clear due to low available data.

## 7. Conclusion

As the research has demonstrated the advantages of using our model to fast detect an ICH case is significantly outweigh the disadvantages which can be needed in a busy Emergency room to get a good idea about the case before final diagnoses from a specialist. Of course, our Research did not reach the optimal results that many other researchers have achieved, and that being said the model could train to get better and get more accurate results. With further research and tuning which can be done in the future by obtaining better data and more research on the subject and trying different approaches to solve this task to get a robust model that can be deployed to have a use in diagnosing ICH in daily medicine life. Many AI researchers believe that AI will be able to do anything humans can do, even in some cases, it can do it better. This is a dubious belief, but AI would undoubtedly outperform humans in some realms. Which is what we are aiming for, by utilizing it to benefit from it in medical diagnoses. I believe that using AI in medical section will save a countless amount of lives that would be otherwise miss diagnosed or would not reach the needed medical help in time. All of this can be resolved by building and deploying such models that can speed up the diagnosing process or even can catch details that humans miss or miss diagnose.



## 8. Resumé

### Úvod

Cieľom tejto práce bolo identifikovať intrakraniálne krvácanie pomocou hlbokého učenia so spoľahlivými výsledkami prechodom „CT skenu“ mozgu do modelu hlbokého učenia, ktorý sme navrhli tak, aby vyhovoval takejto úlohe, a zistiť, či je dané CT sken ICH pozitívny alebo negatívny a zistiť, ktorý podtyp ICH má.

Implementáciou našej úlohy by sme dostali rýchlu diagnostiku, ktorá pomôže urýchliť diagnostický proces na pohotovosti tým, že včas upozorní na nadchádzajúci prípad.

### Hĺbkové učenie

V kapitole „Hlboké učenie“ sme vysvetlili koncept hlbokého učenia a z čoho pozostáva a ako sa použil na zostavenie modelov na klasifikáciu tried od ostatných tried, a vysvetliť, ako je možné vytvoriť architektúry hlbokého učenia pomocou nenásytného prístupu vo vrstvách. Hlboké učenie pomáha pri odbúrání týchto abstrakcií a pri určovaní, ktoré funkcie zvyšujú výsledky.

A spomenuli sme, ako prístupy znižujú funkčné inžinierstvo v učebných činnostiach pod dohľadom tým, že prevádzajú dáta do kompaktných sprostredkovaných reprezentácií podobných hlavným komponentom a generujú vrstvené štruktúry, ktoré eliminujú redundanciu v reprezentácii.

Potom sme vysvetlili základov jeho architektúry a konštrukcie vrstiev a ako tento sa dá použiť na vybudovanie neurónovej siete, ktorá sa neskôr slúži na riešenie skutočných úloh.

## Konvolučné siete

V kapitole „Konvolučné siete“ sme vysvetlili, čo sú to konvolučné siete a ako sú známe aj ako konvolučné neurónové siete alebo CNN, ktoré sú typom neurónovej siete používanej na spracovanie údajov so známou topológiou podobnou mriežke. Dva príklady sú údaje časových radov, ktoré možno považovať za 1-D mriežku vzoriek odobratých v konštantných časových intervaloch, a obrazové údaje, ktoré možno považovať za 2-D mriežku pixelov.

Ďalej vysvetlíme architektúru konvolučnej vrstvy a ako to je tvorená množstvom filtrov, ktorých parametre musia byť známe. Výška a hmotnosť filtrov sú menšie ako je výška vstupného objemu. Každý filter je spojený so vstupným objemom, aby sa vytvorila mapa aktivácie neurónov. Inými slovami, filter sa posúva po šírke a výške vstupu a na každom priestorovom mieste sa počítajú bodové produkty medzi vstupom a filtrom.

A ako Lokálne pripojenie konvolučnej vrstvy umožňuje sieti učiť sa filtre, ktoré sa optimálne prispôbujú miestnej oblasti vstupu, a tak využíva priestorovú miestnu koreláciu vstupu, ktorým je pixel vo vstupnom obraze, viac súvisí so susednými pixelmi ako so vzdialenými pixelmi.

Neskôr sme hovorili o spojovacej vrstve a vysvetlil sme ako Jednou z nevýhod výkonu mapy funkcií konvolučnej vrstvy je, že zaznamenáva presné umiestnenie prvkov na vstupe. To znamená, že aj nepatrné zmeny v umiestnení prvku na vstupnom obrázku vedú k odlišnej mape prvkov. Združovacia vrstva pracuje na každej mape prvkov samostatne a vytvára novú kolekciu klastrovaných máp funkcií s rovnakým počtom prvkov.

Namiesto učenia sa je definovaná operácia združovania. Nasledujú dve tradičné funkcie združovania, priemerné združovanie a maximálne združovanie.

Na výstupnom konci štandardnej hlbokoj neurónovej siete (DNN), ako je napríklad konvolučná neurónová sieť, sa používa úplne spojená vrstva. Konvolučnú neurónovú sieť možno rozdeliť do dvoch etáp, ktorými sú extrakcia a klasifikácia prvkov.

Nasledne sme hovorili o úplne prepojenej vrstve a ako sa dá uvažovať ako o posuvnom klasifikátore, ktorý sa pohybuje okolo diagramu atribútov (Conv). Je to tak, že vo väčšine situácií je prínosom priestorová mierka 1x1. Výsledkom je, že každá konvolučná neurónová sieť je Plne pripojená vrstva.

## Prenos učenia

V kapitole „Transfer learning“ sme si zadefinovali, čo to znamená a ako je to skvelá technika strojového učenia, ktorá nám umožňuje odovzdať model, ktorý bol zostavený pre jednu úlohu, aby sa dal znovu použiť na ďalšie ciele z podobnej domény.

Kľúčovou vlastnosťou, ktorú je potrebné v súvislosti s transferovým učením pochopiť, je to, že transferové učenie sa neobmedzuje iba na oblasť hlbokého učenia, ale to, čo dáva hlbokému učeniu jeho popularitu, sú obrovské a potrebné zdroje na školenie modelov hlbokého učenia alebo v niektorých prípadoch obrovské súbory údajov, na ktorých sa tieto modely trénujú.

Ako už bolo povedané, vymenovali sme niektoré prístupy k prenosu učenia, napríklad „Vytvorte model, ktorý sa má opakovane použiť“, „Používanie vopred vyškoleného modelu“ a „Vytvorte model na extrahovanie funkcií“.

## Súvisiace práce

V kapitole „Súvisiace práce“ boli publikované práce podobné našej úlohe. v každom z týchto výskumných pracovníkov bola klasifikačná úloha implementovaná pomocou mnohých techník, niektoré z nich na riešenie úlohy používali vlastné modely a niektoré z nich používali upravenú verziu vopred vyškolených modelov. V prvej štúdií implementovali vlastné aj vopred pripravené modely vyškolené modely a urobili nejaké porovnanie medzi týmito modelmi a získali podobné výsledky .

Všetky práce mali úlohu detegovať ICH a jeho podtypy sledovaním konkrétného pracovného toku, aby sa k tejto úlohe priblížili.

To znamená, že všetci vedci navrhli rôzne modelové architektúry a implementovali techniky predspracovania do rôznych súborov údajov.

V každej z týchto štúdií sme zhrnuli materiály, ktorý boli používané, typ techník, ktoré použili na predspracovanie údajov, a zhrnili sme typy architektúry, ktoré boli implementované s výsledkami štúdie a určitým porovnaním v prípade rôznych prístupov.

## Naša práca

V kapitole „Naša práca“ sme podrobne opísali náš dátový súbor od spoločnosti Kaggle a vytvorili kompletnú EDA o dátach, aby sme lepšie pochopili, čo máme do činenia. Neskôr sme použili techniky predbežného spracovania, aby sme sa uistili, že náš súbor údajov je štandardizovaný a normalizovaný. Povedali sme, že sme vytvorili tri rôzne okná podľa niektorých hodnôt v „jednotkách Hounsfield“, ktoré sú „subdurálne okná“, „kostné okno“ a „Mäkké okno“, potom sme ich poskladali a vytvorili trojkanálový obraz. Neskôr sme použili dátový vzorkovač na vyváženie súboru údajov, pretože z našej EDA sme si všimli, že súbor údajov má veľa zdravých obrazov mozgu ako tie, ktoré majú ICH, a nechceli sme, aby bol model zaujatý jednou triedou, takže sme vyvážili súbor údajov použitím hybridnej metódy medzi down-samplovaním a nadmerným samplovaním podľa váh tried. Po vyvážení množiny údajov sme zmenili veľkosť množiny údajov,

aby sme ich mohli vložiť do nášho modelu. Na zostavenie nášho modelu sme vytvorili dva modely pre binárnu klasifikáciu, prvý bol vlastný model, ale v tomto modeli sme nedosiahli dobré výsledky, pretože model bol zaseknutý a nebol schopný ďalej trénovať, potom sme sa rozhodli zostaviť model pomocou Transfer learning, a tam sme použili „Resnet-152“ a upravili sme ho tak, že sme zachovali vrstvy extrakcie funkcií a odstránili klasifikačnú vrstvu a pridali vlastnú, čím sme dosiahli výrazne lepšie výsledky, takže sme sa rozhodli použiť rovnakú metódu aj na „Multi-label classification“ „úloha a skutočne sme dosiahli slušné výsledky a dosiahli sme to vytvorením piatich paralelných klasifikačných vrstiev pre každú z tried v našom súbore údajov, ICH podtypy “.

Na záver sme nakreslili niekoľko grafov, ktoré označujú proces učenia s každou stratou tréningu a stratou testu, po ktorom nasledujú výsledky presnosti po každej epoche. potom sme urobili nejaké porovnanie s rôznymi modifikáciami na hyperparametroch aj s vysvetľujúcimi grafmi

## Záver

V závere sme poukázali na to, že naša práca nie je dokončená a je možné k úlohe pridať ďalšiu optimalizáciu s hlbšou analýzou. potom sme spomenuli dôležitosť implementácie techník umelej inteligencie do našich životov, najmä v lekárskom sektore, pretože to môže veľmi pomôcť v tejto oblasti urýchlením diagnostického procesu a dokonca zachytiť niektoré detaily, ktoré my ľudia niekedy nedokážeme zachytiť alebo minúť.

## 9. Resources

- [1] FISCHBEIN, NANCY J. and WIJMAN, CHRISTINE A.C., 2010, Nontraumatic Intracranial Hemorrhage. *Neuroimaging Clinics of North America*. 2010. Vol. 20, no. 4, p. 469-492. DOI 10.1016/j.nic.2010.07.003. Elsevier BV
- [2] HEMPHILL, J. CLAUDE, GREENBERG, STEVEN M. and ANDERSON, CRAIG S., 2015, Guidelines for the Management of Spontaneous Intracerebral Hemorrhage. *Stroke*. 2015. Vol. 46, no. 7, p. 2032-2060. DOI 10.1161/str.0000000000000069. Ovid Technologies (Wolters Kluwer Health)
- [3] NISHIJIMA, DANIEL K., OFFERMAN, STEVEN R. and BALLARD, DUSTIN W., 2013, Risk of Traumatic Intracranial Hemorrhage In Patients With Head Injury and Preinjury Warfarin or Clopidogrel Use. *Academic Emergency Medicine*. 2013. Vol. 20, no. 2, p. 140-145. DOI 10.1111/acem.12074. Wiley
- [4] PURRUCKER, JAN C., HAAS, KIRSTEN and RIZOS, TIMOLAOS, 2016, Early Clinical and Radiological Course, Management, and Outcome of Intracerebral Hemorrhage Related to New Oral Anticoagulants. *JAMA Neurology*. 2016. Vol. 73, no. 2, p. 169. DOI 10.1001/jamaneurol.2015.3682. American Medical Association (AMA)
- [5] CHAN, TAO, 2007, Computer aided detection of small acute intracranial hemorrhage on computer tomography of brain. *Computerized Medical Imaging and Graphics*. 2007. Vol. 31, no. 4-5, p. 285-298. DOI 10.1016/j.compmedimag.2007.02.010. Elsevier BV
- [6] KUO, WEICHENG, HÄNE, CHRISTIAN and MUKHERJEE, PRATIK, 2019, Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning. *Proceedings of the National Academy of Sciences*. 2019. Vol. 116, no. 45, p. 22737-22745. DOI 10.1073/pnas.1908021116. Proceedings of the National Academy of Sciences
- [7] CHANG, P.D., KUOY, E. and GRINBAND, J., 2018, Hybrid 3D/2D Convolutional Neural Network for Hemorrhage Evaluation on Head CT. *American Journal of Neuroradiology*. 2018. Vol. 39, no. 9, p. 1609-1616. DOI 10.3174/ajnr.a5742. American Society of Neuroradiology (ASNR)
- [8] REN, SHAOQING, HE, KAIMING and GIRSHICK, ROSS, 2017, Object Detection Networks on Convolutional Feature Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017. Vol. 39, no. 7, p. 1476-1481. DOI 10.1109/tpami.2016.2601099. Institute of Electrical and Electronics Engineers (IEEE)
- [9] KUO, WEICHENG, HÄNE, CHRISTIAN and MUKHERJEE, PRATIK, 2019, Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning. *Proceedings of the National Academy of Sciences*. 2019. Vol. 116, no. 45, p. 22737-22745. DOI 10.1073/pnas.1908021116. Proceedings of the National Academy of Sciences
- [10] SAGE, AGATA and BADURA, PAWEL, 2020, Intracranial Hemorrhage Detection in Head CT Using Double-Branch Convolutional Neural Network, Support Vector Machine, and Random Forest. *Applied Sciences*. 2020. Vol. 10, no. 21, p. 7577. DOI 10.3390/app10217577. MDPI AG

- [11] BENGIO, YOSHUA, GOODFELLOW, IAN and COURVILLE, AARON, 2016, Deep learning. Massachusetts : MIT Press. ISBN: 0262035618: 326-366
- [12] HUBEL, D. H. and WIESEL, T. N., 1968, Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*. 1968. Vol. 195, no. 1, p. 215-243. DOI 10.1113/jphysiol.1968.sp008455. Wiley
- [13] MARIA FARINELLA., GIOVANNI and MARCO., LEO, 2018, Computer Vision for Assistive Healthcare. Academic Press.
- [14] NAIR, VINOD and HINTON, GEOFFREY, 2010, Rectified Linear Units Improve Restricted Boltzmann Machines. In : . Toronto : Department of Computer Science, University of Toronto. 2010.
- [15] BENGIO, YOSHUA, GOODFELLOW, IAN and COURVILLE, AARON, 2016, Deep learning. Massachusetts : MIT Press. ISBN: 0262035618: 507
- [16] BENGIO, YOSHUA, GOODFELLOW, IAN and COURVILLE, AARON, 2016, Deep learning. Massachusetts : MIT Press. ISBN: 0262035618: 194
- [17] Deep Sparse Rectifier Neural Networks, Xavier Glorot, Antoine Bordes, Yoshua Bengio; *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings* 15:315-323, 2011.
- [18] BENGIO, YOSHUA, GOODFELLOW, IAN and COURVILLE, AARON, 2016, Deep learning. Massachusetts : MIT Press. ISBN: 0262035618: 342
- [19] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
- [20] BENGIO, YOSHUA, GOODFELLOW, IAN and COURVILLE, AARON, 2016, Deep learning. Massachusetts : MIT Press. ISBN: 0262035618: 483
- [21] DENG, LI, 2014, Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*. 2014. Vol. 7, no. 3-4, p. 199-200. DOI 10.1561/20000000039. Now Publishers
- [22] Bengio, Yoshua (2009). "Learning Deep Architectures for AI". *Foundations and Trends in Machine Learning*. 2 (1): 1–127 DOI:10.1561/22000000006
- [23] LECUN, YANN, BENGIO, YOSHUA and HINTON, GEOFFREY, 2015, Deep learning. *Nature*. 2015. Vol. 521, no. 7553, p. 436-444. DOI 10.1038/nature14539. Springer Science and Business Media LLC
- [24] SCHMIDHUBER, JÜRGEN, 2015, Deep learning in neural networks: An overview. *Neural Networks*. 2015. Vol. 61, p. 85-117. DOI 10.1016/j.neunet.2014.09.003. Elsevier BV
- [25] Shigeki, Sugiyama (2019-04-12). *Human Behavior and Another Kind in Consciousness: Emerging Research and Opportunities: Emerging Research and Opportunities*. IGI Global. ISBN 978-1-5225-8218-2.

- [26] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2006. Greedy layer-wise training of deep networks. In Proceedings of the 19th International Conference on Neural Information Processing Systems (NIPS'06). MIT Press, Cambridge, MA, USA, 153–160.
- [27] J. Schmidhuber, "Deep learning", Encyclopedia of Machine Learning and Data Mining, pp. 1-11, 2016.
- [28] DRUMMY, MAX and FULLAN, MICHAEL, 2020, Dive into deep learning. Thousand Oaks, California : Corwin.
- [29] CASTRO, JUAN SEBASTIAN and CHABERT, STEREN, 2021, Convolutional neural networks for detection intracranial hemorrhage in CT images. Ceur-ws.org [online]. 2021. [Accessed 7 May 2021]. Available from: [http://ceur-ws.org/Vol-2564/shortarticle\\_5-CRoNe2019.pdf](http://ceur-ws.org/Vol-2564/shortarticle_5-CRoNe2019.pdf)
- [30] CACERES, J. ALFREDO and GOLDSTEIN, JOSHUA N., 2012, Intracranial Hemorrhage. Emergency Medicine Clinics of North America. 2012. Vol. 30, no. 3, p. 771-794. DOI 10.1016/j.emc.2012.06.003. Elsevier BV
- [31] BRODER, JOSHUA, 2011, Diagnostic imaging for the emergency physician. Philadelphia, PA : Elsevier/Saunders
- [32] MAHESH, MAHADEVAPPA, 2013, The Essential Physics of Medical Imaging, Third Edition. Medical Physics. 2013. Vol. 40, no. 7, p. 077301. DOI 10.1118/1.4811156. Wiley
- [33] TANG, DAVID, 2021, See like a Radiologist with Systematic Windowing. Kaggle.com [online]. 2021. [Accessed 7 May 2021]. Available from: <https://www.kaggle.com/dcstang/see-like-a-radiologist-with-systematic-windowing>
- [34] YE, HAI and GAO, FENG, 2019, Precise diagnosis of intracranial hemorrhage and subtypes using a three-dimensional joint convolutional and recurrent neural network. European Radiology. 2019. Vol. 29, no. 11, p. 6191-6201. DOI 10.1007/s00330-019-06163-2. Springer Science and Business Media LLC
- [35] BENGIO, YOSHUA, GOODFELLOW, IAN and COURVILLE, AARON, 2016, Deep learning. Massachusetts: MIT Press. ISBN: 0262035618: 526
- [36] SORIA OLIVAS, EMILIO, 2010, Handbook of research on machine learning applications and trends. Hershey, PA : Information Science Reference.
- [37] MORRIS, TIM, 2004, Computer vision and image processing. Basingstoke : Palgrave Macmillan. ISBN 978-0-333-99451-1
- [38] DANG, L. MINH and MIN, KYUNGBOK, 2020, Tampered and Computer-Generated Face Images Identification Based on Deep Learning. Applied Sciences. 2020. Vol. 10, no. 2, p. 505. DOI 10.3390/app10020505. MDPI AG
- [39] HE, KAIMING and ZHANG, XIANGYU, 2016, Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. DOI 10.1109/cvpr.2016.90. IEEE
- [40] DONGES, NIKLAS, 2020, WHAT IS TRANSFER LEARNING? EXPLORING THE POPULAR DEEP LEARNING APPROACH. BuiltIn.com [online]. 2021. [Accessed 7 May 2021]. Available from: <https://builtin.com/data-science/transfer-learning>



[41] Radiological Society of North America, RSNA Intracranial Hemorrhage Detection Identify acute intracranial hemorrhage and its subtypes, kaggle.com [online]. [Accessed 7 May 2021]. Available from: <https://www.kaggle.com/c/rsna-intracranial-hemorrhage-detection/data>