# An Efficient CNN-BiLSTM Model for Multi-class Intracranial Hemorrhage Classification

**Kevin Genereux**

Electrical and Computer Engineering

Lakehead University, Thunder Bay, Ontario

A thesis submitted to Lakehead University in partial fulfillment
of the requirements for the Master of Science degree
in the Electrical and Computer Engineering

# Examining Committee Membership

The following served on the Examining Committee for this thesis:

| | |
|---|---|
| Supervisor: | Dr. Thangarajah Akilan<br>Assistant Professor, Department of Software Engineering. |
| Internal Committee Member: | Dr. Hassan Naser<br>Associate Professor, Department of Software Engineering. |
| External Committee Member: | Dr. Garima Bajwa<br>Assistant Professor, Department of Computer Science. |
| Session Chair: | Dr. Yushi Zhou<br>Associate Professor, Department of Electrical and Computer Engineering. |

# Declaration of Co-Authorship / Previous Publications

## I. Co-Authorship Declaration

I hereby declare that this dissertation incorporates material that is the result of joint research, as follows: This dissertation incorporates the outcome of research under the supervision of Dr. T. Akilan. In all cases, the key ideas, primary contributions, experimental designs, data analysis, interpretation, and writing were performed by the author, and the contribution of the coauthors was primarily through the provision of proofreading and reviewing the research papers regarding the technical content.

I am aware of the Lakehead University Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my dissertation and have obtained permission from each co-author to include the above materials in my dissertation.

I certify that, with the above qualification, this dissertation, and the research to which it refers, is the product of my work.

# II. Declaration of Previous Publications

This thesis includes the following conference paper that was accepted for publication.

| Thesis chapter | Publication title/full citation | Status |
|---|---|---|
| Chapter 4 | K. Genereux and T. Akilan, An Efficient CNN-BiLSTM-based Model for Multi-class Intracranial Hemorrhage Classification, International Conference on Image Vision and Computing (ICIVC), 2023 | Accepted and Presented |

# III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act. I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office. This thesis has not been submitted for a higher degree to any other University or Institution.

# Acknowledgements

I would like to take this opportunity to express my deepest gratitude to all those who have supported and guided me throughout the journey of completing this thesis. Without their invaluable contributions, this accomplishment would not have been possible.

First and foremost, I am extremely grateful to my supervisor, Dr. Thangarajah Akilan, for his unwavering support, mentorship, and guidance throughout the entire research process. His expertise, patience, and insightful feedback have been instrumental in shaping this thesis and my growth as a researcher. I am truly fortunate to have had the privilege of working under his supervision.

I would like to extend my heartfelt appreciation to the members of my thesis committee, Dr. Hassan Naser, Dr. Garima Bajwa, and Dr. Yushi Zhou, for their valuable insights, constructive criticism, and scholarly expertise. Their meticulous review of my work and thoughtful suggestions have greatly enhanced the quality and rigor of this thesis.

I am indebted to my family for their unconditional love, unwavering support, and understanding throughout this academic journey. Their belief in me and constant encouragement have been my driving force during challenging times.

# Dedication

*This thesis is dedicated to my dear parents.*

# Abstract

Intracranial hemorrhage (ICH) refers to a type of bleeding that occurs within the skull. ICH may be caused by a wide range of pathology, including, trauma, hypertension, cerebral amyloid angiopathy, and cerebral aneurysms. Different subtypes of ICH exist based on their location in the brain, including epidural hemorrhage (EDH), subdural hemorrhage (SDH), subarachnoid hemorrhage (SAH), intraventricular hemorrhage (IVH), and intraparenchymal hemorrhage (IPH). Prompt detection and management of ICH are crucial as it is a life-threatening medical emergency with high morbidity and mortality rates. Despite accounting for only 10-15% of all strokes, ICH is responsible for over 50% of stroke-related deaths. Therefore, the presence, type, and location of an ICH must be immediately diagnosed so that the patients can receive medical intervention. However, accurately identifying ICH in CT slices can be challenging due to the brain's complex anatomy and the variability in hemorrhage appearance.

Recent advancements in deep learning models have demonstrated their effectiveness in interpreting medical images, often surpassing the performance of trained medical specialists. This research presents an efficient deep learning approach that combines a 2-D convolutional neural network (CNN) with a bidirectional long-short-term memory (Bi-LSTM) module to achieve accurate ICH detection and subtype classification. Notably, this study introduces the integration of a multi-head attention mechanism into the CNN-BiLSTM design, which improves performance for this task. The experimental results on three publicly available benchmark datasets demonstrate the system's high performance and strong generalizability. The inclusion of the multi-head attention mechanism within the proposed CNN-BiLSTM effectively reduces the weighted multi-label binary cross-entropy with logits loss score, from 0.0501 to 0.0482 on the RSNA dataset. Moreover, the proposed solution exhibits competitive results on the CQ500 dataset, yielding an

accuracy of 0.959, sensitivity of 0.974, specificity of 0.958, and precision of 0.977. It achieves AUC scores of 0.9869, 0.9797, and 0.9778 on the RSNA, CQ500, and PhysioNet datasets, respectively. The proposed solution holds the potential to be deployed as an intelligent assistive tool for radiologists, aiding them in accurately and efficiently diagnosing ICH. However, it is imperative to recognize the potential challenges associated with deploying the proposed solution in a clinical setting. Clinical environments often feature variations in CT scanner acquisition parameters and patient demographics that may differ from the data used for model training. Consequently, there is a possibility that our solution's performance may experience some degradation when applied in these real-world clinical scenarios.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| Acronym | Description |
|---|---|
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| AVM | Arteriovenous Malformation |
| AUC | Area under the ROC Curve |
| BA | Bat Algorithm |
| BiLSTM | Bidirectional Long Short-term Memory |
| BPH | Brain Polytrauma Hemorrhage |
| CNN | Convolutional Neural Network |
| CT | Computed Tomography |
| DL | Deep Learning |
| DRLSE | Distance Regularized Level Set Evolution |
| EDH | Epidural Hemorrhage |
| ELM | Extreme Learning Machine |
| EHO | Elephant Herd Optimization |
| FCM | Fuzzy C-Means Clustering |
| FCN | Fully Convolutional Network |
| FN | False Negative |
| FP | False Positive |
| GC | GrabCut |

| Acronym | Description |
|---|---|
| GLCM | Gray Level Occurrence Matrix |
| GLRLM | Gray Level Run Length Matrix |
| GNN | Graph Neural Network |
| GWO | Grey Wolf Optimization |
| GOA | Grasshopper Optimization Algorithm |
| GRU | Gated Recurrent Unit |
| Grad-CAM | Gradient-weighted Class Activation Mapping |
| GPU | Graphics Processing Unit |
| HPC | High-Performance Computing |
| HU | Hounsfield Unit |
| IAT | Intelligent Assistive tool |
| ICH | Intracranial Hemorrhage |
| IVH | Intraventricular Hemorrhage |
| KNN | K-Nearest Neighbors |
| LASSO | Least Absolute Shrinkage and Selection Operator |
| Leaky ReLU | Leaky Rectified Linear Unit |
| LSTM | Long Short-term Memory |
| ML | Machine Learning |
| MLP | Multi-layer Perceptron |
| MDRLSE | Modified Distance Regularized Level Set Evolution |
| MRI | Magnetic Resonance Imaging |
| MML | Minimalist Machine Learning |
| NLP | Natural Language Processing |
| NN | Neural Network |
| PCA | Principal Component Analysis |
| PACS | Picture Archiving and Communication System |

| Acronym | Description |
|---|---|
| RF | Random Forest |
| RNN | Recurrent Neural Network |
| ROC | Receiver Operating Characteristic Curve |
| ROI | Region of Interest |
| RSNA | Radiological Society of North America |
| SAH | Subarachnoid Hemorrhage |
| SDL | Synergistic Deep Learning |
| SDH | Subdural Hemorrhage |
| SVM | Support Vector Machine |
| TP | True Positive |
| TN | True Negative |
| ViT | Vision Transformer |
| WL | Window Level |
| WW | Window Width |
| WOA | Whale Optimization Algorithm |
| 2-D | Two Dimensional |
| 3-D | Three Dimensional |

# Chapter 1

# Introduction

## 1.1 Background of Intracranial Hemorrhage

Intracranial hemorrhage (ICH) is a serious medical condition characterized by bleeding within the brain. It is a medical emergency that necessitates immediate attention due to its potential to cause severe neurological complications and even death if not treated promptly [5, 6]. ICH is life-threatening because blood accumulation within the skull increases pressure on the brain. This elevated pressure can compress brain tissue and interfere with its normal function which can cause various neurological deficits. Furthermore, the bleeding associated with ICH disrupts the delivery of essential oxygen and nutrients to brain cells, leading to tissue damage or cell death. The consequences of this deprivation can have lasting effects on a person's cognitive, motor, and sensory abilities. ICH can be caused by multiple factors, including head injuries, hypertension (high blood pressure), blood clotting disorders, and ruptured aneurysms [6]. Certain risk factors increase the likelihood of experiencing ICH, such as advanced age, high blood pressure, smoking, alcohol or drug abuse, and blood clotting disorders [7]. The symptoms of ICH can vary depending on the location and severity of the bleeding. Common signs include a severe headache, confusion, nausea, vomiting, seizures, weakness or numbness in the limbs, difficulty speaking or understanding speech, and loss of consciousness [8]. Diagnosing ICH typically involves a comprehensive approach. It includes evaluating the patient's medical history, conducting a thorough physical

examination, and utilizing advanced imaging tests such as computed tomography (CT) scans or magnetic resonance imaging (MRI) of the brain. These diagnostic techniques enable healthcare professionals to visualize the extent and location of the bleeding to assist in the development of an appropriate treatment plan [6]. Treatment for ICH often requires surgical intervention aimed at removing the blood clot, repairing damaged blood vessels, or alleviating pressure on the brain [9]. It is crucial to emphasize the importance of prompt diagnosis and treatment of ICH. Rapid medical intervention can help limit the extent of bleeding, relieve pressure on the brain, and minimize subsequent neurological impairments [10]. Early treatment significantly reduces the risk of permanent brain damage or death. Rehabilitation and supportive care are often essential components of the recovery process for individuals with ICH. These measures assist patients in regaining lost functions and adapting to any ongoing difficulties resulting from the initial bleeding [9].

## 1.2 Motivation

Radiology is currently facing a critical need for the development of deep learning (DL) algorithms specifically designed to detect ICH. A DL algorithm for detecting ICH can serve as an Intelligent Assistive Tool (IAT) that augments a radiologist's abilities to diagnose ICH with improved accuracy and efficiency. The consequences of misdiagnosing ICH and failing to provide immediate medical intervention can be severe for patients. Studies have revealed that although ICH represents only 10-15% of all strokes, it is responsible for more than 50% of stroke-related deaths overall [2, 11]. Moreover, the mortality rate within 30 days of an ICH event ranges from 35% to 52%, with only 20% of survivors experiencing a complete recovery within six months of symptom onset [12, 13]. In fact, more than one-third of survivors are left with severe disabilities three months after their diagnosis [14]. These alarming statistics emphasize the need for an IAT that can enhance diagnostic accuracy and ultimately improve the chances of a successful recovery.

DL algorithms have the potential to analyze extensive medical data and uncover complex patterns and correlations that may not be readily apparent to trained medical professionals. Timely and accurate diagnosis of ICH has significant implications for patient outcomes and healthcare re-

source utilization. DL algorithms can rapidly analyze and prioritize CT scans, flagging cases with suspected hemorrhages for immediate attention. Before the advent of deep learning, traditional images preprocessing techniques were used for medical imaging analysis. These methods involved applying filters, edge detection, morphological operations, and region-growing approaches (refer to Chapter 2 for more details). However, traditional images preprocessing techniques generally are not as flexible and generalizable as DL approaches in detecting complex, irregularly shaped hemorrhagic regions. The rapid identification of ICH empowers healthcare professionals to make informed decisions regarding surgical evacuation and interventions aimed at preventing further complications. Through streamlining the diagnostic process, an IAT has the potential to elevate patient care, reduce healthcare costs, and ultimately improve patient outcomes for individuals affected by ICH [15].

Additionally, the development of an IAT for diagnosing ICH can alleviate the growing burden on radiologists, who are faced with increasingly demanding workloads. As CT scanners continue to advance in sophistication, they generate higher-resolution CT scans, resulting in a larger number of images per scan. Moreover, there has been a consistent annual growth in the number of patients receiving CT scans. Consequently, radiologists are tasked with the challenge of analyzing larger volumes of data, which can lead to fatigue and an increased likelihood of misdiagnoses. To meet the rising demand, it has been estimated that a radiologist would need to examine each CT slice within approximately 3 seconds while working an 8-hour shift. Such an overwhelming workload can result in fatigue and interpretation errors, including misdiagnosis or missed diagnosis. From this perspective, an IAT can provide invaluable assistance to radiologists by initially flagging potential abnormalities in CT scans and prioritizing critical cases that require immediate attention [14].

Once developed, DL algorithms for detecting ICH have the potential to be scaled and deployed across different healthcare facilities. This scalability allows for the widespread implementation of advanced diagnostic tools, including in remote regions where access to specialized medical resources may be limited. By providing healthcare facilities with access to DL algorithms, patients in remote regions can benefit from an increased level of diagnostic accuracy as those in more

developed areas. This equitable access to radiology diagnostic tests empowers healthcare providers to make informed decisions regarding the diagnosis and treatment of patients with suspected ICH, regardless of their geographical location [16].

## 1.3 CT Scans

**Figure 1.1:** Illustration of a patient undergoing a head CT scan [1].

Radiologists rely on CT scans as a primary diagnostic tool for detecting ICH. The procedure begins with the patient lying on a doughnut-shaped machine known as a CT scanner, as depicted in Fig. 1.1. This advanced technology utilizes X-ray beams projected through the patient's head at various angles, with detectors measuring the amount of radiation that passes through the tissues. The collected data is then processed by a computer, which reconstructs the information into detailed cross-sectional images of the brain [17]. Radiologists use specialized software to enhance the contrast of the CT scans and then review the resulting CT images. They meticulously examine the entire series of images, also referred to as CT scan slices, in order to identify any abnormal findings. Specifically, radiologists focus on detecting areas of abnormal density, as blood appears hyperdense in comparison to the surrounding brain tissue on CT scans. By carefully evaluating the density and characteristics of the images, radiologists can distinguish between different subtypes of hemorrhages, aiding in accurate diagnosis and appropriate treatment planning.

Additionally, radiologists assess the CT images for associated findings beyond the hemorrhage itself. These may include midline shift, which refers to the displacement of brain structures due to the accumulating blood. They also look for signs of mass effect, which involves the compression of surrounding structures caused by the hemorrhage. Furthermore, radiologists evaluate the images for signs of increased intracranial pressure, skull fractures, or other secondary complications related to the bleeding. This comprehensive evaluation allows them to provide a thorough assessment of the patient's condition [18, 19].

Hounsfield Units (HU) are a measurement scale used in CT scans to quantify the radiodensity of tissues and structures within the body. When X-rays pass through different tissues and structures, they are attenuated based on their interaction with the materials encountered. The attenuation is determined by the atomic composition and density of the tissue or structure. HUs represent this attenuation and are derived from the linear attenuation coefficient, which measures the reduction in X-ray intensity as it passes through a particular material. Water, having a density of 1 $g/cm^3$, was assigned a HU value of 0. HUs are calculated relative to this reference point, with positive values assigned to denser materials and negative values assigned to less dense materials. Air, for example, has a HU value of around -1000 due to its low density, while dense bones can have values greater than +1000. HU values are also important in identifying pathologies and abnormalities [17, 20]. For instance, ICH can typically be identified by a narrow range of HU values between 60 to 100 [21].

CT scans play a crucial role in assisting medical professionals with diagnosing ICH due to their high sensitivity, accessibility, rapid results, detailed imaging capabilities, and relative safety. Firstly, they are highly sensitive to the presence of blood and can accurately detect even small hemorrhages. Blood appears as hyperdense regions on CT scans, allowing radiologists to identify ICH. Furthermore, radiologists can utilize the visual information obtained from CT scans to determine the subtype of the hemorrhage. Another advantage of CT scanning is its widespread availability in hospitals, along with the relatively quick procedure it entails. This accessibility and efficiency ensure that CT scans provide rapid results, facilitating prompt decision-making for further intervention when necessary. In terms of imaging capabilities, CT scans provide detailed

cross-sectional images of the brain and surrounding structures, including the different lobes, ventricles, and structures within the skull. This comprehensive coverage is crucial for accurately detecting and evaluating the precise location and extent of the hemorrhage. CT scans are generally considered safe, as they do not involve the use of ionizing radiation during the imaging process. This characteristic makes them suitable for a wide range of demographics, including children and pregnant women, when necessary [22].

## 1.4 ICH Subtypes

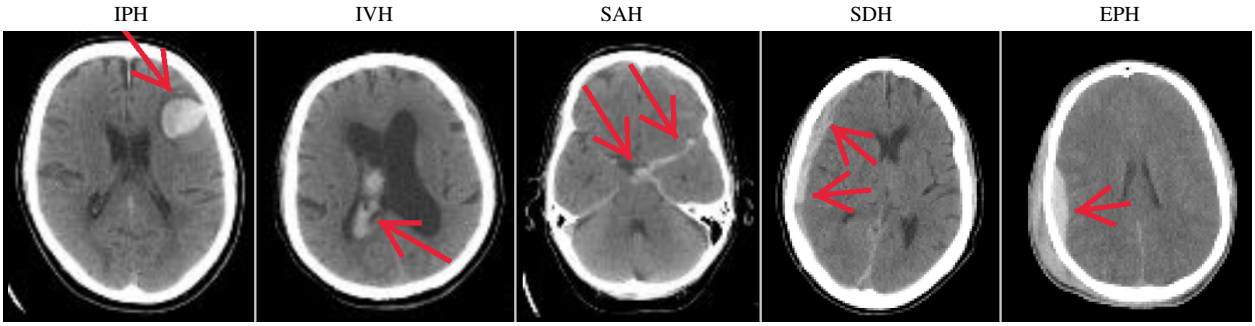



**Figure 1.2:** ICH subtypes: intraparenchymal (IPH), intraventricular (IVH), subarachnoid (SAH), subdural (SDH), and epidural hemorrhage (EDH). The locations of these subtypes are as follows. IPH - inside the brain, IVH - inside the ventricle, SAH - between the arachnoid and the pia mater, SDH - between the dura and the arachnoid, and EDH - between the dura and the skull [2].

Depending on the specific location where the bleeding occurs, ICH can be classified into five distinct subtypes: intraparenchymal hemorrhage (IPH), intraventricular hemorrhage (IVH), subarachnoid hemorrhage (SAH), subdural hematoma (SDH), and epidural hematoma (EDH). Each subtype presents unique characteristics in terms of location. In this section, a broad overview of the subtypes of ICH is provided. Fig. 1.2 provides a depiction of each ICH subtype.

### 1.4.1 IPH

IPH is a bleed that occurs within the brain parenchyma, the functional tissue in the brain consisting of neurons and glial cells. There are various factors that can contribute to this type of

hemorrhage, including but not limited to hypertension, arteriovenous malformation, amyloid angiopathy, aneurysm rupture, tumor, coagulopathy, infection, vasculitis, and trauma. IPH refers to bleeding that occurs within the actual brain tissue [23].

### 1.4.2 IVH

IVH is a condition characterized by bleeding that occurs inside or around the ventricles of the brain, which are fluid-filled spaces containing cerebrospinal fluid. IVH can have various causes, but it is often secondary to other underlying conditions such as trauma, aneurysms, vascular malformations, or tumors. In many cases, IVH occurs as a result of the expansion of an existing IPH or SAH, where bleeding extends into the ventricles.

IVH is more commonly observed in premature babies or those with very low birth weight. In infants, the bleeding typically takes place in a region called the germinal matrix, while in full-term babies, it originates from the choroid plexus. The occurrence of IVH in premature infants is generally not attributed to physical injury but rather to changes in blood flow and the delicate structures of the developing brain. The immature circulatory system of the brain in premature babies is more susceptible to oxygen deprivation, which contributes to the development of IVH. Reduced blood flow can lead to cell death in the brain and rupture of blood vessel walls, resulting in bleeding [24].

### 1.4.3 SAH

SAH is the most frequently encountered type of ICH resulting from trauma. SAH refers to bleeding into the space between the brain's surface and the thin membrane covering it, called the subarachnoid space. Subarachnoid hemorrhages are frequently caused by the rupture of an aneurysm (a weakened and bulging blood vessel) or an arteriovenous malformation (an abnormal tangle of blood vessels). SAH can be further classified as aneurysmal and non-aneurysmal SAH. Aneurysmal SAH happens when a cerebral aneurysm ruptures, causing bleeding into the subarachnoid space. On the other hand, non-aneurysmal subarachnoid hemorrhage refers to bleeding into the

subarachnoid space without any identifiable aneurysms. Non-aneurysmal subarachnoid hemorrhage is most frequently associated with trauma, specifically blunt head injuries, with or without penetrating trauma, or sudden acceleration changes to the head.

The primary cause of SAH is typically trauma, which results in injury to the blood vessels on the surface of the brain, leading to bleeding into the subarachnoid space. On the other hand, non-traumatic SAH most commonly occurs due to the rupture of a cerebral aneurysm. When an aneurysm ruptures, blood is able to flow into the subarachnoid space. Additionally, SAH can be caused by other factors such as arteriovenous malformations (AVMs), the use of blood thinners, trauma, or idiopathic reasons [24].

### 1.4.4 SDH

SDH involves bleeding between the pia and the arachnoid membrane, which is the middle layer covering the brain. SDH happen beneath the protective layer called the dura mater, so their size is not restricted by the skull sutures. This means that SDH can extend across the lines where the bones of the skull connect, which can help differentiate them from EDH. In terms of their dimensions, SDH are contained between the falx cerebri (a membrane in the middle of the brain) from the center towards the sides and the tentorium cerebri (a membrane that separates the cerebrum from the cerebellum) from top to bottom. These SDH often spread along the same side as these structures without any opposing force. It often occurs due to trauma that causes tearing of the veins bridging the brain surface and the dura mater. When SDH is small in size, they can be difficult to identify on a CT scan due to blending with nearby bones [25].

The main cause of SDH is typically damage to the veins that are located deep beneath the dura mater. Unlike EDH, SDH are less likely to be accompanied by skull fractures [25–27]. Due to the slower rate of growth caused by venous injury and the larger lateral space available in the subdural area, SDH is often not immediately symptomatic compared to EDH. However, when SDHs become large, they can exert significant pressure on the nearby brain, potentially leading to midline or transtentorial herniation. In such cases, emergency evacuation of the hematoma may be necessary [25].

### 1.4.5 EDH

This type of hemorrhage occurs when blood accumulates between the skull and the outermost protective layer of the brain called the dura mater [25]. EDH can originate either from an artery or a vein. The classical arterial EDH arises as a result of blunt trauma to the head, often affecting the temporal region. It can also manifest following a penetrating injury to the head. In these cases, a fracture of the skull is typically observed, which leads to damage in the middle meningeal artery and subsequent arterial bleeding into the potential epidural space. In the case of a venous EDH, it occurs when there is a fracture in the skull, and the bleeding from this fracture fills the space between the skull and the dura mater. Venous EDH is frequently observed in pediatric patients [24].

In severe cases, the blood spreads along the inner part of the skull until it reaches the closest line where the skull bones are connected, called a suture. When the bleeding reaches this point, it starts to expand sideways along the suture line. At the same time, the bleeding also becomes thicker from the surface towards the deeper part, creating a shape that looks like a curved or rounded crescent. This is why we call these types of bleeding inside the head ”crescentic” or ”biconvex” EDH [24].

EDH can be identified clinically by a period of time called a ”lucid interval,” during which the patient may not show severe symptoms or be critically ill. This interval occurs between the traumatic injury and the buildup of bleeding in the epidural space, before it starts to put pressure on the nearby brain. However, as the EDH grows larger, it can exert significant pressure on the surrounding brain, leading to midline, subfalcine, or trans-tentorial herniation. This can cause a rapid decline in the patient's level of consciousness or even result in death [24, 25].

## 1.5 Challenges

The difficulty for radiologists to detect ICH in CT scans can be attributed to several factors.

- **Varying appearance**: The appearance of ICH can vary significantly in terms of morphology, including shape, size, and location. In some cases, ICH may coexist with other intracranial

pathologies, such as tumors, abscesses, or ischemic strokes. This variability in ICH appearance can complicate the interpretation of CT scans and make it more challenging to isolate and identify the hemorrhage accurately [28].

- **Low contrast difference**: Calcifications, artifacts, and normal anatomical structures can create similar shades of gray on CT scans, making it harder to differentiate ICHs from these confounding factors. Radiologists must carefully examine each image and consider multiple features, such as density, shape, location, and surrounding edema, to distinguish ICH from other structures accurately [18].

- **Image artifacts and noise**: CT images can be affected by various artifacts and noise, such as motion artifacts, beam hardening artifacts, and metal artifacts. These can obscure or mimic the appearance of ICH, leading to diagnostic challenges [25].

- **Varying clinical settings**: Different hospitals may employ various CT scanner types and imaging protocols, leading to variations in image quality and acquisition techniques.

- **Time constraints**: In emergency situations, prompt and accurate identification of ICH is crucial for initiating appropriate interventions. However, the limited time available for image interpretation and the pressure to make rapid decisions can increase the likelihood of overlooking or misinterpreting subtle signs of hemorrhage [14].

## 1.6 Objectives

This study seeks to develop a sophisticated DL solution for the identification and classification of ICH that emulates the interpretation process of expert radiologists. The primary objective is to create a highly accurate and efficient model capable of detecting ICH and classifying its subtypes. Ultimately, this DL solution aims to serve as an IAT for radiologists. The integration of this model as an IAT can empower radiologists with improved diagnostics capabilities, thereby enabling faster and more accurate decisions in patient management.

## 1.7 Deep Learning for Medical Imaging

Medical imaging plays a crucial role in healthcare by providing valuable information for the diagnosis, treatment, and monitoring of various medical conditions. It encompasses a wide range of imaging modalities such as X-ray, computed tomography, magnetic resonance imaging, ultrasound, and positron emission tomography (PET). These imaging techniques allow healthcare professionals to visualize internal structures, organs, tissues, and physiological processes. Through medical imaging, medical experts can identify abnormalities, make accurate diagnoses and plan appropriate treatments.

DL has emerged as a powerful approach for automatically extracting meaningful features from complex data, including medical images. Traditionally, medical imaging analysis required meticulous manual effort and expertise from medical specialists to examine images and identify intricate patterns to make accurate diagnoses. However, medical specialists are susceptible to fatigue, distractions, and cognitive biases that can contribute to misdiagnoses or missed diagnoses. However, DL algorithms have revolutionized this approach. By training on extensive labeled medical imaging datasets, these algorithms can discern complex patterns and relationships inherent in the images. Consequently, they are capable of automatically detecting and classifying abnormalities with remarkable speed and accuracy. DL algorithms offer a consistent and objective analysis of medical images that minimize discrepancies in diagnoses.

DL algorithms have shown exceptional performance in several key areas of medical imaging, including classification, segmentation, and radiomics. Classification is an important task in medical imaging where DL algorithms categorize images into specific classes. For example, DL algorithms can classify an image as normal or abnormal. Segmentation involves the delineation of structures or regions of interest within medical images. DL algorithms excel at segmenting organs, tissues, lesions, and other anatomical structures from complex imaging data. For instance, DL-based segmentation methods can accurately outline tumors in radiological images, aiding in treatment planning and monitoring. By precisely delineating regions of interest, DL algorithms facilitate quantitative analysis and assist clinicians in making informed decisions. Radiomics refers

to the extraction and analysis of a large number of quantitative features from medical images. DL algorithms can automatically extract intricate features from images, such as texture, shape, and intensity, which may be challenging for human observers. These features can then be used to build predictive models or identify imaging biomarkers associated with specific diseases or treatment outcomes. DL-powered radiomics has the potential to enhance precision medicine by providing personalized insights into disease progression [29, 30].

Several key factors have contributed to the rapid rise of DL in the field of medical imaging. Large datasets are generally required for the successful training of DL models. In recent years, there has been a significant increase in the availability of annotated medical image datasets due to advancements in digital storage, improved data-sharing practices, and collaborative efforts by researchers and medical institutions. These large datasets enable DL models to learn complex patterns and generalize well. In addition, the availability of high-performance Graphics Processing Units (GPUs) and parallel computing architectures has made it feasible to train DL models on large-scale medical image datasets. The improved computing power reduces the training time and allows researchers to develop and optimize models more efficiently. Advances in computational power have enabled researchers to explore increasingly sophisticated DL models to extract intricate patterns from medical images.

## 1.8 Technical approach

To address the objectives mentioned above, a DL solution is designed to automatically identify ICH. In Chapter 4, a DL model is employed that utilizes a windowing technique as an image preprocessing step. This technique enhances the contrast of each CT scan slice, allowing for better detection of subtle abnormalities associated with ICH. By applying an extensive set of data augmentations to the CT scan slices, we aim to increase the diversity and variability of the training data. This augmentation process improves the generalization and robustness of the trained models.

To learn distinct feature representations of ICH, a 2-D Convolutional Neural Network (CNN) model is trained. Subsequently, a bidirectional Long Short-Term Memory (BiLSTM) network is

utilized to capture sequential patterns among the CT scan slices. The BiLSTM network takes the feature embeddings generated by the CNN and leverages them to extract temporal dependencies and long-range interactions. This integration of a CNN and BiLSTM allows the model to effectively analyze the sequential nature of CT scan data, leading to improved predictive accuracy.

For evaluation purposes, the trained model generates multi-label predictions for each sample in the RSNA-ICH test set, CQ500, and PhysioNet-ICH datasets. We measure the performance of the model using key metrics such as sensitivity, specificity, precision, and the Area Under the Curve (AUC) score. These metrics provide a comprehensive assessment of the model's ability to correctly identify instances of ICH.

In Chapter 5, a multi-head attention mechanism is introduced into the model to learn global dependencies and capture long-range interactions within the sequential feature embeddings. This attention mechanism enables the model to focus on relevant features and emphasize the most informative regions within the CT scan slices. By attending to these important regions, the proposed model can better discern patterns associated with ICH, leading to enhanced predictive accuracy and performance.

## 1.9 Thesis contribution

This study's primary contributions are as follows:

- It proposes an efficient CNN-BiLSTM model harnessing a pre-trained EfficientNetV2-Small for primitive feature detection and a shallow sequence learning model for capturing the CT scan's slice-dependent cues (Chapter 4).

- It applies pragmatic regularization measures to address the overfitting problem in which a model begins to memorize the training dataset instead of learning underlying patterns. An extensive set of data augmentations is applied to each CT scan slice in the training dataset. This effectively increases the diversity and variability of the training data, thereby improving generalization ability (Section 4.1.1 - Data Preprocessing).

- It incorporates countermeasures to address the class imbalance problem in the benchmark datasets. To accomplish this, the vanilla binary cross-entropy loss function described in Section 4.1.4 is modified to place more emphasis on identifying ICH subtypes since they are underrepresented in the datasets. Precision and the area under a Receiver Operating Characteristic (ROC) curve are employed for evaluation metrics to ensure that the proposed model is effective at classifying samples from minority classes (Section 4.2.2 - Evaluation Metrics).

- It introduces the integration of a multi-head attention mechanism into the CNN-BiLSTM design, which improves the predictive accuracy of the model while incurring a trivial increase in inference time (Section 5.2 - Proposed Solution).

# Chapter 2

# Related Works

The related works section offers a thorough examination of existing research and literature that are pertinent to the topic. It delves into common image processing techniques that were used to enhance the quality of CT scan slices and eliminate unwanted noise. Furthermore, it offers an analysis of various studies used for the classification and segmentation of ICH and its subtypes. This section aims to present a balanced assessment of the strengths and limitations inherent in existing research approaches. Additionally, it puts forth potential improvements to address the identified limitations. The insights gained from this review have the potential to support future research endeavors and foster the development of precise and efficient diagnostic tools for detecting ICH. Ultimately, these advancements contribute to the enhancement of patient care and outcomes.

## 2.1 Preprocessing

Preprocessing is an essential step in improving the quality of CT scan slices by eliminating potentially unnecessary elements, including the skull, CT scan rest, edema, and background. This information is typically irrelevant and may hinder image analysis algorithms from being able to focus on the pertinent features required for identifying ICH [31].

Thresholding is a common image preprocessing technique that has been used for detecting ICH that converts a grayscale CT scan slice into a binary format. Image thresholding separates an image into two classes by determining a threshold value, where pixels above the threshold are considered the intracranial region. By highlighting the intracranial region, subsequent image analysis algorithms may be able to identify intracranial bleeding more accurately and efficiently [32–39].

CT scans often contain noise due to factors like equipment limitations, patient movement, and small metallic objects. Noise can interfere with the accuracy of detecting hemorrhagic regions. To reduce noise, Chan used a median filter to replace each pixel's value with the median value of its neighboring pixels. This technique helps to preserve edges and details while smoothing out random noise [37].

Although CT scans are typically acquired in anisotropic resolution, they can be transformed into isotropic resolution to improve spatial coherency. A CT scan with anisotropic resolution has a non-uniform spatial resolution with varying voxel dimensions between CT slices. In contrast, isotropic resolution refers to a uniform spatial resolution. Anisotropic resolution can result in varying voxel dimensions between CT slices, which can hinder the ability of a model to learn spatial relationships. Converting to isotropic resolution ensures consistent voxel size throughout the entire CT scan and enables models to better understand spatial relationships [34, 40].

Morphological operations, namely erosion and dilation, have been used for refining and enhancing the ICH regions. Erosion is beneficial in ICH detection for removing small artifacts and fine structures while leaving behind the larger and more prominent ICH regions. In contrast, dilatation is used to bridge the small breaks in hemorrhagic regions to ensure the completeness and

continuity of the ICH. These small breaks may lead to a fragmented detection and inaccurate representation of the ICH size and shape if left unaddressed [32, 37, 41–44].

Inspired by the conventional radiology workflow, windowing has been used as a preprocessing technique to enhance the contrast levels of the CT scan. The process involves manipulating the grayscale values of pixels within a specified range of interest to enhance the visibility of specific anatomical structures or pathologies. This is achieved by assigning lower and upper thresholds, known as window levels and window widths, which determine the range of pixel values to be mapped to new display values. This approach highlights subtle discrepancies in pixel intensity between ICH and surrounding tissues, which makes it easier to identify ICH [19, 45–50].

Data augmentations have been used by various studies to artificially increase the size and diversity of the training dataset for detecting ICH. This is accomplished by applying geometric image transformations to introduce variations into the training data without the need for additional data collection. This process ensures that the model learns more representative ICH features from the training dataset. For example, rotation can be used to help the model learn to recognize ICH from different orientations. As a result, the models become more adept at handling variations in CT scan slices [34, 35, 38, 39, 51].

## 2.2 Classification

This section examines classification models used to accurately classify ICH and their subtypes. By training on labeled examples, classification models learn patterns from CT scan slices to effectively assign the correct label to each instance.

### 2.2.1 Traditional Methods

Alawad *et al.* extracted a comprehensive set of 17 features from the region of interest (ROI), such as the size of the ROI, centroid of the ROI, perimeter of the ROI, and the distance between the ROI and the skull. To enhance performance, they employed a genetic algorithm (GA)-based feature selection technique to identify the most relevant features. The selected features were then

utilized to train a stacking-based ML framework, comprising Support Vector Machine (SVM), Random Decision Forest (RDF), Extra Tree (ET), K-Nearest Neighbors (KNN), Bagging (BAG), and Logistic Regression(LR). Among these classifiers, the SVM model using a radial basis function (RBF) demonstrated the best performance. To evaluate the effectiveness of the model, 10-fold cross-validation was employed, yielding results with an accuracy of 0.995, precision of 0.99, recall of 0.989, and F1 score of 0.989. However, it is worth noting that the study did not include an independent test to assess the model's performance on unseen data. It is crucial to validate the model on an independent test to avoid potential bias in the training data and provide a more reliable assessment of the model's effectiveness and generalizability [32].

Raghavendra *et al.* generated texture features using Gray Level Occurrence Matrix (GLCM), Gray Level Run Length Matrix (GLRLM), and Hu moments. In addition, pixel-intensity features are extracted using first-order statistics, such as kurtosis, skewness, and variance. Nature-inspired meta-heuristics algorithms, including the Bat Algorithm (BA), Grey Wolf Optimization (GWO), and Whale Optimization Algorithm (WOA), are used to select the best set of features. To address the class imbalance, Adaptive Synthetic Sampling (ADASYN) is used to create synthetic ICH samples since ICH is underrepresented in the training dataset. The selected features were then used to train a KNN classifier for ICH subtype classification. Overall, the findings demonstrate the effectiveness of meta-heuristics algorithms for feature selection in the application of ICH [52].

After obtaining the ROI using Otsu's thresholding method, Al-Ayoob *et al*. extracted discriminating features about the ROI, including, its size, centroid, perimeter, distance between the ROI and the skull, and the eccentricity of an ellipse having the same second-moments as the ROI. These features were used for training a multinomial LR classifier to distinguish between EDH, SDH, and IPH. However, their study only includes a limited dataset of 76 CT scans. Due to the relatively low sample size, their model may not be generalizable to account for variations in other clinical settings. In comparison, the CNN-BiLSTM used in this thesis was trained on a large scale dataset and was then tested on two external datasets to demonstrate its generalizability [33].

Zaki *et al.* employed multi-level FCM to extract the intracranial region from the background and skull. Then, a two-level Otsu multithresholding method is employed to segment the intracra-

nial structure into cerebrospinal fluid, brain matter, and other homogeneous areas. By using the symmetrical properties within the intracranial structures, a quantitative comparison is performed between the segmented left-half and right-half regions with respect to the intracranial midline. This analysis is valuable for identifying normal and abnormal structures within the intracranial region, as any detected asymmetry suggests a high likelihood of abnormalities. Additional information derived from the pixel intensities, such as the standard deviation and maximum value within the segmented regions, is used to differentiate between ICH and normal cases [53].

### 2.2.2 Deep Learning Methods

Recently, CNNs have emerged as the predominant approach for detecting ICH in CT scans. CNNs can learn to recognize complex patterns in visual data by training on a large number of annotated samples and are excellent in handling variability in contrast, rotation, and lighting conditions [54, 55].

Several studies have employed pre-trained CNN architectures for detecting ICH and have applied transfer learning. Transfer learning is a powerful technique that allows the knowledge acquired during training for one task (source task) to be utilized for a different, but related task (target task). In the context of ICH detection, transfer learning enables the model to benefit from prior learning on general image recognition tasks, leading to quicker convergence and potentially improved generalization. The model can understand essential image features, edges, textures, and patterns, which are transferable and useful for ICH detection.

Chilkamurthy *et al.* proposed using the ResNet18 architecture with five parallel fully connected layers as the output layers. The output from these layers for each slice was fed to a Random Forest (RF) model for classification. The model underwent training using an extensive dataset consisting of 290,000 CT scan slices. Additionally, a subset of 21,000 CT scan slices was used specifically for the validation process. The training and validation CT scans were labeled using a natural language processing (NLP) algorithm, with the clinical radiology reports serving as the trusted reference standard. While the model achieved an average sensitivity of 92%, it only obtained a specificity of 70%. The model performance can potentially be improved by incorporating data augmentations

to help the CNN with learning different representations of ICH. In comparison, this thesis uses an extensive set of eleven different augmentations to help prevent the model from overfitting to noise in the training set [19].

Kumaravel *et al.* has exploited AlexNet to extract high-level features from CT scan slices. To address the class imbalance problem, the number of CT scan slices without ICH was undersampled to match the number of CT scan slices with ICH. Principal Component Analysis (PCA) is applied to reduce the dimensionality of the feature space. Finally, the reduced features are fed into an Support Vector Machine (SVM) classifier, which learns to identify ICH based on the extracted features. While their model achieves a binary classification accuracy of 0.9986 in detecting ICH, their model was not designed for multiple ICH subtype classifications, which may limit its clinical utility. In clinical practice, ICH subtype diagnosis is important for determining appropriate medical intervention strategies [36].

Ngo et al *et al.* implemented a CNN using the ResNet-50 architecture with a two-stage process. During training, each slice is sampled with three axial slices before and after it to generate output descriptors. In the second stage, the output descriptors are used as input into a 3-layer CNN, containing only two convolutional layers and one FC layer, for generating confidence scores for the center slice. However, sampling only the three axial slices before and after each slice during training may limit the model's ability to understand long-term sequential dependencies among CT scan slices. ICH can exhibit complex patterns and variations that may extend beyond the immediate neighboring slices. In addition, this process is more computationally expensive than a conventional 2-D CNN model as it involves extracting features from seven adjacent CT scan slices to generate a confidence score for the center slice.

Ensemble models are beneficial because they combine the predictions of multiple models, resulting in improved accuracy and robustness. For example, He *et al.* developed an ensemble of 2-D CNN models built using the SE-ResNeXt50 and EfficientNet-B3 architectures for detecting ICH. This study includes random cropping and label smoothing to improve model robustness [56]. Lee *et al.* presented an ensemble model that combines VGG16, ResNet-50, Inception-v3, and Inception-ResNet-v2. Their approach involves using attention maps to enhance the reliability

of the localization process and a prediction basis that provides a comprehensive explanation for the model's predictions [46]. By leveraging the strengths of different models and reducing biases, ensembles provide more reliable predictions and handle complex problems more effectively. On the other hand, ensemble models have increased complexity and require additional computational resources.

CT scans are inherently volumetric and can be represented by a stack of a series of 2-D axial slices. Studies have explored using 3-D CNNs to learn features across the entire volume and learn complex spatiotemporal patterns among adjacent slices. Ker *et al.* apply pixel-intensity thresholding to improve the performance of their 3-D CNN model for classifying SAH, IPH, SDH, and brain polytrauma hemorrhage (BPH). A dataset containing 399 CT scans with approximately 12,000 CT slices from the National Neuroscience Institute at Tan Tock Seng Hospital was used for training and evaluating the model. A Gaussian distribution was used to initialize the 3-D convolutional kernels, while parameter tuning was performed using SGD and a cross-entropy loss function. Without any thresholding, The F1 scores for different pairs of medical diagnoses in a 2-class classification scenario varied between 0.706 and 0.902. Upon applying thresholding, the F1 scores improved and ranged from 0.919 to 0.952 [35].

Likewise, Titano developed a 3-D CNN based on the ResNet-50 architecture for the purpose of triaging patients exhibiting acute neurological events. Remarkably, in a simulated clinical setting, the model performs inference about 150 times faster for a CT scan compared to a radiologist (1.2 seconds versus 177 seconds). A potential performance improvement would be to apply windowing to enhance the contrast of the CT scans before being fed into the model. In this thesis, a combination of brain, subdural, and soft tissue windows were used to help the model identify potential abnormalities [40].

Researchers have also investigated the fusion of 2-D CNN models with a sequential learning model. This approach is less computationally burdensome than a 3-D CNN while still enabling the model to learn complex spatiotemporal patterns among adjacent slices.

For instance, Alis *et al*. developed a joint CNN-RNN model with an interspersed attention layer to capture the most relevant data. A large-scale dataset of 55,179 head CT scans was retrospec-

tively collected from five different centers. After development, the DL model was integrated into a center's Picture Archiving and Communication System (PACS) environment for performance assessment in a clinical setting. During the prospective implementation, the model yielded an accuracy of 96.02% on 452 head CT scans with an average accuracy of 45 $\pm$ 8 s. The study proposed their novel NormGrad technique to generate saliency maps. NormGrad aims to highlight the decision-making process of a model by calculating the outer product between vectorized components of activation maps and gradients using the Frobenius Norm. Using the Mann-Whitney-U test, Norm-Grad was shown to produce higher-quality saliency maps than the Gradient-weighted Class Activation Mapping (Grad-CAM) method [47].

Yeo et al. also used a CNN-RNN approach for ICH detection and subtype classification. The model was trained on the Radiological Society of North America (RSNA) Brain CT Hemorrhage dataset which contains 752,803 CT scan slices. The CQ500 dataset was used as an independent dataset for testing and verifying the generalizability and robustness of the trained model in handling variations of different clinical settings. Two different image pre-processing pipelines were considered. In the first approach, each CT scan slice was converted into three contrast-enhanced channels using brain subdural, and soft tissue windows. The second approach involved a slice concatenation technique where each CT slice was concatenated with the CT slice immediately before and after it to also create a three-channel input. Each image preprocessing technique was used for training a separate CNN-RNN model and the results were ensembled. To improve the interpretability of their model, Grad-Cam visualizations were produced that show which regions in a CT scan slice were most relevant in generating the final predictions. A fixed learning rate was used for training the model. To further improve performance, a learning rate scheduler can be used to help the model converge faster and help prevent the model from becoming stuck on a suboptimal solution [50].

However, it is worth noting that RNNs face difficulty with capturing long-term dependencies due to the vanishing gradient problem. During training, the gradients can diminish or explode as they propagate through many time steps, making it challenging for the network to effectively

capture and remember information from distant past inputs. This limitation can hinder the RNN's ability to model sequences with long-term dependencies accurately.

Kadam *et al.* developed a framework consisting of three different models: Xception, Xception-LSTM, and Xception-GRU. Their comparative analysis reports that the Xception-GRU achieves the lowest log loss score while being marginally more computationally efficient than the Xception-LSTM. The model achieved a specificity of 0.9915 but only obtained a sensitivity of 0.7317. To reduce this discrepancy in sensitivity and specificity performance, the underrepresented ICH subtype classes can be oversampled so that the model is better able to learn discriminative features of ICH [12].

In an early DL-based study, Grewal *et al.* created a baseline network that employed a modified 40-layer DenseNet architecture with three dense blocks. By adding three auxiliary tasks to the network to focus on hemorrhagic regions, Grewal *et al.* was able to improve the classification performance. The auxiliary task branches were implemented after the final concatenation layer in each dense block, where each branch was made up of a single module comprising a $1 \times 1$ convolutional layer followed by a deconvolution layer. This design enabled the feature maps to be upsampled to their original image size. The upsampled output is used as input into a global pooling layer, passed through a BiLSTM, and then sent to an FC layer for generating final predictions. The proposed model achieved higher recall than two of the three radiologists [34].

Nguyen *et al.* have also developed a CNN-BiLSTM for ICH classification. Although our approach has some similarities, it differs from their method in the following aspects. This thesis incorporates dropout ratios in the sequential learning subnetwork to mitigate overfitting issues during training. In addition, the architecture used in this thesis is more efficient as it only uses 21.5 million trainable parameters in comparison to the SE-ResNeXt-50 implemented by Nguyen *et al.*, which requires 27.6 million. Their work uses brain, subdural, and bone windows, whereas this work uses brain, subdural, and soft tissue windows. Our approach explores incorporating a multi-head attention mechanism into the CNN-BiLSTM design to improve its performance.

On the other hand, Barhoumi and Ghulam propose a hybrid architecture called Scopeformer, which combines multiple CNNs and a multi-encoder vision transformer (ViT) with self-attention

mechanisms. Each CT scan slice is concurrently analyzed by multiple CNN models to extract representative feature maps. These feature maps are then subjected to a patch extraction module, which converts them into vectors. The resultant vectors serve as the input for the multi-encoder ViT to learn rich global feature correlations. An ablation study is conducted to assess how varying the number of CNN modules, convolutional layers, and attention heads impacts the accuracy and computational complexity. However, their best model achieves a weighted multi-label logarithmic loss score of 0.0705, which is comparably lower performance than the state-of-the-art studies [49].

## 2.3 Segmentation

This section examines segmentation models used to identify and delineate the boundaries of ICH in CT scans by learning patterns from labelled examples.

### 2.3.1 Traditional Methods

Traditional approaches use handcrafted features, such as intensity-based or texture-based features, to identify ICH in CT scans.

For instance, Chan proposed using thresholding and morphological operations to segment intracranial contents, followed by denoising and adjustment for cupping artifacts. The intracranial region is then automatically aligned into a conventional position based on the midsagittal plane. High attenuation components were then identified as candidate ICH regions using a top-hat transformation and determining the intensity difference between the two contralateral anatomical regions. To provide anatomical context, the potential AIH regions are aligned with a standardized coordinate system. Finally, a knowledge-based classification system is applied that uses quantified imaging features and anatomical information [37]. However, it should be noted that the effectiveness of the symmetry analysis heavily relies on the assumption that axial slices are perfectly parallel to the axial plane [57].

Fuzzy C-Means clustering (FCM) has commonly been applied to create an initial contour to extract the ROI. Unlike traditional clustering methods, it allows for soft assignments, meaning each

data point can belong to multiple clusters with varying degrees of membership. The algorithm calculates the membership values for each data point based on the similarity to cluster centroids and updates the centroids iteratively until convergence.

For instance, Bhadauria *et al.* designed an integrated approach combining FCM and region-based active contour methods, where FCM clustering is used to initialize the contour and estimate propagation controlling parameters adaptively. In this approach, FCM is used to initialize a contour surrounding the hemorrhagic region. It then employed a region-based active contour method to propagate the initial contour toward the boundaries of the hemorrhage. Additionally, FCM clustering is employed to dynamically estimate the contour propagation parameters based on the given Ct scan slice. Unlike traditional methods, this approach utilizes local intensity information rather than global information to guide the contour motion [41].

In contrast, Gautam *et al.* proposed a novel variant of FCM termed modified robust fuzzy c-means clustering (MRFCM). A hyper tangent function is used as the kernel for fuzzy clustering while distance regularized level set evolution (DRLSE) is the edge-based active contour method for the segmentation of brain hemorrhagic lesions from CT scans. In addition, Lagrange's multiplier is used to calculate the final objective function of the method. The experimental results indicated better performance compared to standard FCM, spatial FCM, robust kernel-based fuzzy clustering (RFCM), and DRLSE algorithms [42].

Kumar *et al*. proposed using FCM to divide each CT scan slice into three different clusters that have similar types of membership values. An automatic selection process is employed to choose a cluster from the three clusters by comparing the skull histogram of the CT scan slice with the FCM clustered images. The method incorporates entropy-based thresholding to further segment the ROI and removes spurious blobs using morphological operations. Then, DRLSE is used by placing an initial contour within the hemorrhagic region for further refinement [58].

Li *et al.* developed a supervised learning approach for segmenting SAH by using the probability of distance features from five anatomical landmarks: brain boundary, midsagittal plane, anterior and posterior intersection points of brain boundary with the midsaggital plane, and superior point of the brain. After extracting landmarks from CT scan slices, the distances to these landmarks for each

pixel in the CT scan slice were calculated. The prior probabilities of distances for pixels considered SAH and non-SAH were computed. Based on the distance features, a Bayesian framework was used to delineate SAH. In contrast to elastic registration that relies on grayscale information, their approach demonstrated reduced susceptibility to grayscale discrepancies observed between normal individuals and patients [43].

Farzaneh *et al.* developed a sophisticated RF model for the binary segmentation of SDH using an extensive set of handcrafted features and DL-derived features. Since the most severe case of SDH were not deeper than 3.2 cm from the skull, the ROI was defined as the intracranial region within 3.2 cm of the inner skull. A level-set method was employed to further segment the ROI enclosed by the skull in case there were any openings in the skull boundary. Once the ROI was established, superpixels were generated using the simple linear iterative clustering (SLIC) algorithm to reduce redundant information. An extensive set of feature extraction was performed to derive patterns about the superpixels. This process included extracting location-based features, histogram-based features, and filtering-based pixels by convolving Gabor and Laplacian of Gaussian filters with the images. A U-Net architecture was employed to obtain data-driven deep features. The obtained features for each superpixel were classified as hematoma or non-hematoma using an RF forest model. Post-processing involved using a 3-D Gaussian smoothing kernel to smooth jagged contours and increase spatial coherency [57].

Yao *et al.* also employed the SLIC algorithm to produce superpixels and then extracted a comprehensive set of features based on pixel-intensity statistics, Gabor filters, saliency, GLCM, and wavelet packet transformation. To address the limited annotated data, an active learning strategy was used to select the most informative unlabeled data for annotation, which was then used to train a SVM classifier. The coarse segmentation from the classifier was further enhanced by incorporating it into an active contour model, resulting in improved segmentation accuracy. The proposed ICH segmentation system proved to be robust in handling patient cases from multiple health centers and multiple levels of injury [59].

### 2.3.2 Deep Learning Methods

Deep learning approaches have commonly employed variations of a U-Net architecture for the segmentation of ICH [44, 60–62]. It consists of a contracting path to capture context and a symmetric expansive path for precise localization. The contracting path performs downsampling using convolution and pooling layers. The expansive path upsamples the features and combines them with the contracting path's corresponding feature maps through skip connections to produce pixel-level segmentation masks [63].

Barros *et al*. developed a shallow CNN model for detecting SAH in a collection of CT scans which was then further evaluated on rebleed patients. Image thresholding was used to segment the bone and then morphological dilation was applied to close all openings in the skull except for the foramen magnum. The centroid of the segmented bone is then used as a seed for the region growing within the skull. Once the ROI was obtained, the preprocessed data was used as input into a primitive 2D U-Net model with two convolutional and two FC layers. The model was also trained based on randomly sampled background patches. Despite the frequent occurrence of severe metal artifacts in the scans of rebleed patients caused by coiling, the CNN-based segmentation method seems to be appropriate for accurately segmenting rebleeds as well, with similar accuracy. On average, the CNN detection and segmentation process requires 30 seconds per CT scan [60].

Hssayeni *et al*. developed a fully automated U-Net model for the segmentation of ICH from 82 CT scans. However, the authors report a Dice coefficient score of 0.31, which is comparatively low to state-of-the-art methods. The authors note that the model is biased towards false positive segmentation, especially in regions near the bones where the intensity in grayscale values is similar to that of ICH regions. To alleviate this issue, pixel intensity thresholding can be explored to remove the bone regions from the CT scans and mitigate this bias. Alternatively, the U-Net can be modified by incorporating an attention mechanism to help it focus on relevant regions during the encoding and decoding stages. The main contribution of their work is that they made their PhysioNet-ICH dataset publicly available with 82 CT scans to enable further research studies [44].

In contrast, Wang *et al.* developed a semi-supervised, attention-based U-Net. An inverse sigmoid-based curriculum learning training strategy was employed to stabilize the training process.

Their model significantly outperforms the conventional U-Net described in [44] as it achieves a Dice coefficient of 0.67 while also using the PhysioNet-ICH dataset. However, their model suffered from the class imbalance problem as it was less effective at segmenting smaller ICH regions compared to larger ones. To mitigate this issue, a different loss function, such as Tversky loss, could be used that places additional emphasis on identifying regions with ICH [61].

Yao *et al.* introduced dilated convolutions into their modified U-Net model and removed down-sampling layers. The addition of dilated convolutions allowed the model to effectively capture and analyze information from a broader range of spatial scales. This is particularly advantageous for the task of segmenting ICH as they tend to vary considerably in both size and shape. To improve generalization capabilities, an L2 weight decay regularization of $4 \times 10^{-4}$ was used. The model was very effective in segmenting very large hematomas with a Dice coefficient score of 0.80 but only obtained a Dice coefficient score of 0.59 at classifying small hematoma [64].

Kuo *et al.* introduced PatchFCN, a fully convolutional network (FCN) that utilizes a patch-based approach to achieve accurate segmentation and categorization of hematomas. An FCN is trained on random small patches cropped from whole images centered in the foreground. Using small patches of a CT scan slice instead of the entire CT scan slice enabled higher batch sizes that better represent the diversity of the dataset. However, their model does not consider the spatial relationships between adjacent CT scan slices. To address this limitation, future studies may consider incorporating a sequential model, such as an LSTM, to learn these spatial dependencies and enhance predictive accuracy [39].

Cho suggested utilizing a deep learning model composed of two CNNs and two fully convolutional networks in a cascade format for the segmentation and classification of SAH, EDH, and SDH. Their large-scale training dataset included 135,974 CT scan slices The two CNNs are employed to categorize the existence of ICH. In case ICH is detected, the two FCN models are utilized for subtype classification and segmentation. Each of the CNN/FCN models underwent separate training, utilizing two distinct window settings, namely the default CT scan window setting and a stroke window. An overall segmentation performance of 80.19% precision and 82.15% recall for

delineating bleeding lesions, demonstrating a 3.44% improvement compared to the utilization of a single FCN model [65].

Chang *et al.* proposed a custom hybrid 3D/2D mask R-CNN architecture for ICH evaluation. A preconfigured distribution of bounding boxes with distinct resolutions and shapes is initially used. Based on the scores assigned to each bounding box, high-ranking ones were selected for generating focused region proposals. Non-maximal suppression was used to prune the composite region proposals before being used as input into a classifier to ascertain hemorrhage absence or presence. In cases where the classification result is positive, a binary mask was produced by the final segmentation network branch [38].

Ertugrul and Akil employed a YOLO-V4 network that creates a bounding box to segment the ICH and then perform classification. Class label smoothing was used as a regularization technique to reduce overconfidence by distributing probability mass across incorrect labels. While a bounding box may provide a rough approximation of an ICH's location and shape, it cannot accurately capture the precise contours, orientations, and scale of an ICH. As a result, the use of bounding boxes may lead to inaccuracies in localization and decrease model performance [51].

# Chapter 3

# Overview of Neural Networks and Attention

This chapter focuses on providing a foundational understanding of four DL models: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) network, and Attention. The knowledge presented in this chapter forms the basis for Chapters 4 and 5, where the DL techniques will be applied and evaluated for their effectiveness in ICH detection and classification.

The chapter begins with an overview of ANNs, which serve as the building blocks for many DL architectures, such as CNNs and LSTMs. It discusses the basic principles underlying ANNs, including multi-layer perceptron networks, activation functions, and the training process. Next, the chapter delves into CNNs, which are designed to extract spatial and hierarchical representations from input data. This portion explains the intricate architecture of CNNs, including convolutional layers, pooling layers, and fully connected layers. The subsequent section explores LSTM networks, a type of RNN known for their ability to capture temporal dependencies in sequential data. Lastly, attention mechanisms are introduced as a mechanism to enhance the performance of deep learning models by focusing on relevant regions or features within an image. The section introduces self-attention, a mechanism that allows a model to weigh the importance of different elements in a sequence based on their relevance. Furthermore, it explores multi-head attention, an extension of self-attention that enables the model to attend to multiple subspaces simultaneously.

## 3.1 Artificial Neural Network

### 3.1.1 Overview

ANNs are a computational model inspired by the structure and functioning of biological neural networks, such as the human brain. Fig. 3.1 presents a simple illustration of the structure of a biological neuron. The structure and function of a neuron in an ANN closely resemble that of a neuron in a biological neural network. Both types of neurons have a similar organization, consisting of input connections, a processing unit, and an output connection. In biological neurons, signals are received through dendrites which are akin to the input connections in artificial neurons. The cell body integrates these signals, and if a threshold is exceeded, an action potential is generated. This is similar to the output signal or activation in an artificial neuron. Both types of neurons use an activation function to determine their output. This function depends on factors such as membrane potential and synaptic strength in biological neurons. In both cases, synaptic connections are vital for information transfer. Biological networks can change connection strengths through synaptic plasticities, such as long-term potentiation or depression. In contrast, artificial networks adjust input importance using weights and learning algorithms like backpropagation. Although ANNs are simplified models inspired by biological networks, they aim to approximate their behavior [3, 66].

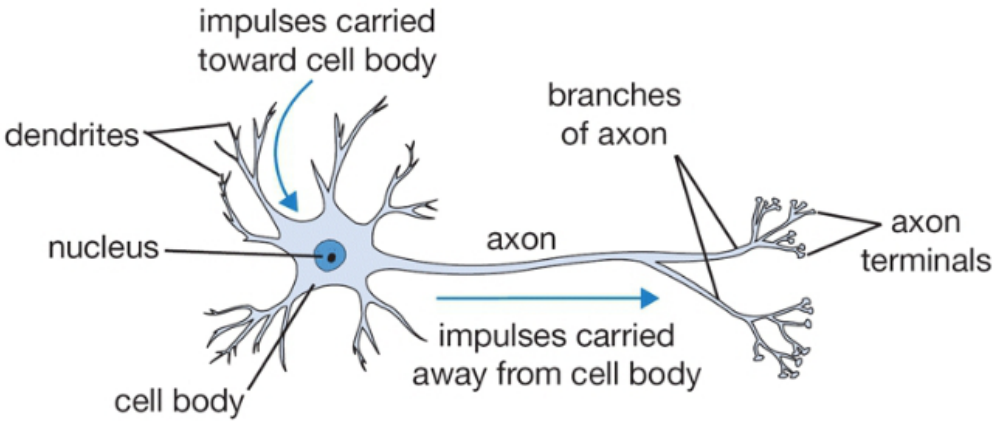



**Figure 3.1:** A visual depiction of a biological neuron [3].

### 3.1.2 Multi-Layer Perceptron

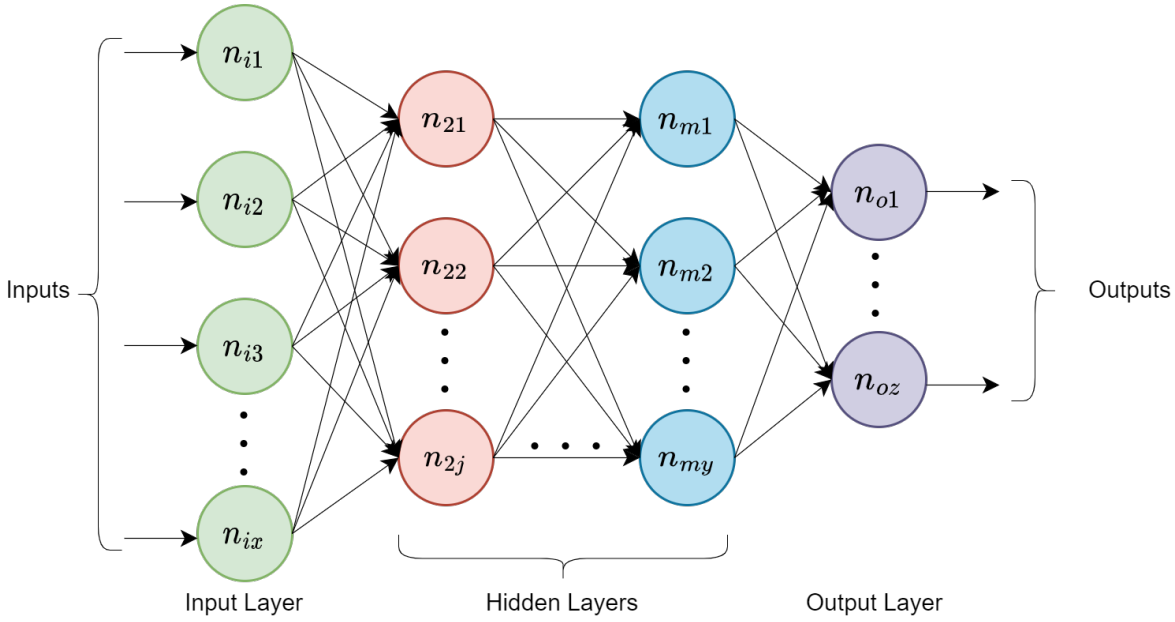



**Figure 3.2:** A general illustration of an MLP network that consists of interconnected layers of neurons. The network contains an input layer denoted by $i$, one or more hidden layers denoted by $m$, and an output layer denoted by $o$. $n$ represents a neuron. The number of neurons in the input, hidden, and output layers is denoted by $x$, $j$, $y$, and $z$, respectively.

A multi-layer perceptron (MLP) neural network is a type of ANN that consists of multiple layers of interconnected artificial neurons, also known as perceptrons. Fig. 3.2 presents a general illustration of the structure of an MLP network. An MLP is a feedforward neural network as the information flows in one direction from the input layer through the hidden layers to the output layer. The input layer consists of artificial neurons, each representing a feature of the input data. The number of neurons in the input layer is equal to the number of input features. MLPs can have one or more hidden layers, which are layers between the input and output layers. Each hidden layer contains artificial neurons that transform the input using weighted connections and activation functions. The output layer consists of artificial neurons that generate the final predictions or outputs. Each artificial neuron in one layer is connected to every neuron in the subsequent layer. Each connection between two neurons has a weight associated with it, which determines the strength or importance of the connection.

The fundamental component of an ANN is the artificial neuron. The artificial neuron takes multiple input signals, each of which is multiplied by a corresponding weight value. The weighted inputs are summed together. The weights are iteratively adjusted during the training process to enhance the network's performance. The weighted summation operation for an artificial neuron is computed as follows in (3.1).

$$z = \sum_{i=1}^{N} (x_i \cdot w_i) + b, \tag{3.1}$$

where $z$ represents the output value, $N$ is the number of elements in the input vector $x$ and weights vector $w$, $b$ is a bias term, and $i$ is the index.

The weighted sum is then passed through an activation function, which introduces non-linearities into the neuron's response. The activation function determines whether the neuron should "fire" or activate, based on the aggregated input. The output of an artificial neuron is the result of the activation function applied to the weighted sum [67].

### 3.1.3 Activation Functions

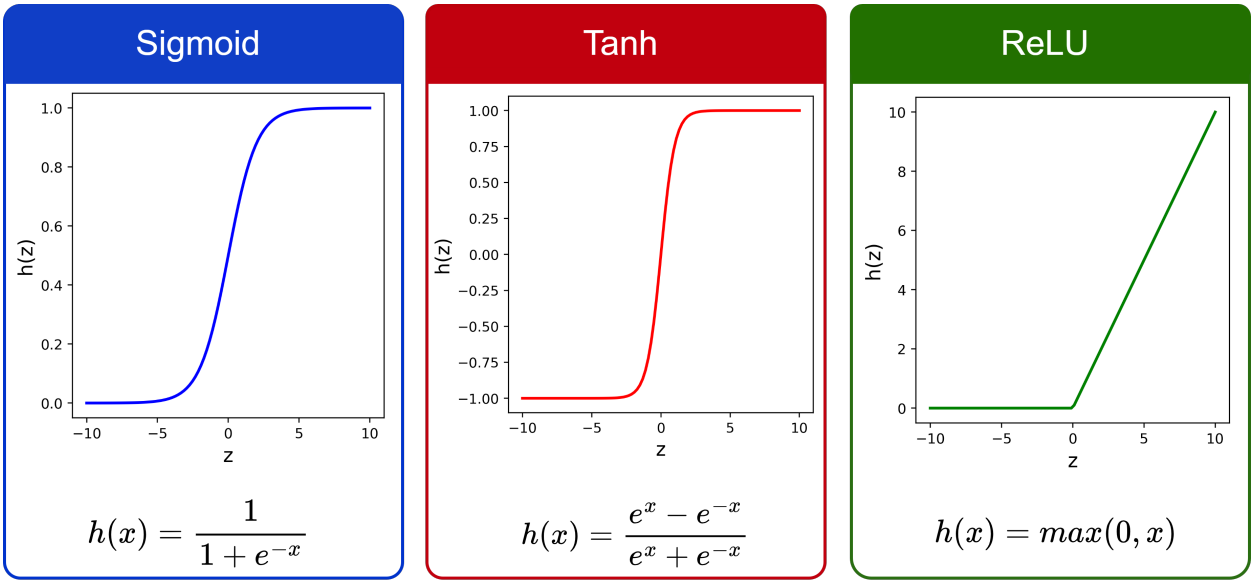



**Figure 3.3:** Common activations functions employed in ANNs.

Activation functions are a crucial component of ANNs as they introduce non-linearity into the network. Non-linearity is crucial in ANNs because it enables the network to capture complex relationships and patterns within the data that would be otherwise impossible to represent using only linear transformations. In an MLP, activation functions are commonly employed to learn and model complex relationships between inputs and outputs. Some commonly used activation functions used in MLPs include sigmoid, ReLU, tanh, and softmax. Fig. 3.3 shows the graphs of common activation functions.

**Sigmoid:** The sigmoid function is a classic activation function that maps any input value to a value between 0 and 1. The sigmoid function is mathematically expressed as (3.2):

$$h(x) = \frac{1}{1 + e^{-x}}, \tag{3.2}$$

where $x$ is the input to the function. It has a smooth and continuous curve that gradually transitions from 0 to 1 as the input value increases.

**Tanh:** The hyperbolic tangent (tanh) function maps the input value to a range between -1 and 1. This property makes it suitable for tasks that require symmetric activation around zero. The tanh function is mathematically expressed as (3.3):

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}, \tag{3.3}$$

where $x$ is the input to the function.

**ReLU:** The Rectified Linear Unit (ReLU) function has gained popularity in recent years due to its simplicity and effectiveness in deep learning models. ReLU is a piecewise linear function which only activates when the input value is positive, which allows for faster learning in deep networks. ReLU helps mitigates the vanishing gradient problem by maintaining a constant input of 1 for positive inputs. This issue tends to arise in traditional activation functions like sigmoid or tanh when gradients become very small for extreme input values. ReLU's mathematical simplicity and mitigation of the vanishing gradient problem translate to accelerated convergence and consequently reduced training times. Mathematically, the ReLU function can be represented as (3.4):

$$F(x) = max(0, x), \tag{3.4}$$

where $x$ is the input to the function.

**Softmax:** The softmax function is a popular activation function used in the output layer of neural networks for multi-class classification problems. It converts a vector of real numbers into a probability distribution over multiple classes, with each class representing a distinct category and receiving a probability value between 0 and 1. The softmax function is defined as (3.5):

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K \tag{3.5}$$

where $z_i$ denotes the output of the i-th neuron, $K$ is the number of classes, and $j$ represents each element of the input vector.

The softmax function exponentiates the input values and normalizes them by dividing each exponentiated value by the sum of all exponentiated values. This normalization ensures that the output probabilities add up to 1, enabling them to represent the likelihood of the input belonging to each class [67].

### 3.1.4 Training Strategy

Training ANNs involves the process of adjusting the weights of the connections between artificial neurons to optimize the network's performance. During training, a dataset consisting of input samples and their corresponding desired outputs is provided to the neural network. In forward propagation, the input samples are propagated through the network, layer by layer, until the output layer is reached. The output generated by the network is compared to the desired output using a loss function. For example, cross-entropy loss is a commonly employed loss function for classification tasks. The operation for cross-entropy loss is computed as follows in (3.6).

$$E = -\sum_{c=1}^{C} y_c \log(p_c), \tag{3.6}$$

where $E$ represents the cross entropy loss, $y_c$ is the true probability or label for class $c$, and $p_c$ is the predicted probability for class $c$. The sum is taken over all classes $C$.

The goal of training is to minimize the error between the network's output and the desired output. This is achieved by iteratively adjusting the weights in the network based on the calculated error, which is referred to as backpropagation. The key idea behind backpropagation is to propagate this error backward through the network, layer by layer, and update the weights accordingly.

Gradient descent is an iterative optimization algorithm used for updating the weights of a neural network during the training process. The fundamental concept underlying gradient descent involves calculating the gradient of the loss function concerning the weights and subsequently adjusting the weights in a manner that minimizes the loss. The gradient descent algorithm for updating the weights of the network is summarized as follows:

**Step 1 - Initialize Weights:** All weights in the network are initialized with random values.

**Step 2 - Forward Propagation:** The input data is fed into the neural network, and the activations of each layer are computed sequentially through weighted sums and activation functions. The activations of one layer serve as the input to the next layer, and this process continues until the final layer, which produces the network's output.

**Step 3 - Loss Calculation:** The error between the network's output and the desired output is calculated using a suitable loss function.

**Step 4 - Gradients Calculation:** The gradient of the loss function with respect to each parameter is calculated. The gradient represents the direction and magnitude of the steepest ascent of the function. It indicates how much the loss function will change if we change the parameters.

**Step 5 - Weight Update:** The weights and biases are updated using the gradient information and the learning rate. The new weights are calculated by subtracting the product of the learning rate and the gradient from the current weights. This operation for updating the weights can be mathematically expressed as in (3.7).

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} - \eta \frac{\partial L}{\partial w_{ij}}, \tag{3.7}$$

where $w_{ij}^{\text{new}}$ represents the updated weight connecting neuron $j$ in layer $l-1$ to neuron $i$ in layer $l$, $w_{ij}^{\text{old}}$ is the previous weight, $\eta$ denotes the learning rate, $\frac{\partial L}{\partial w_{ij}}$ is the partial derivative of the loss function $L$ with respect to the weight $w_{ij}$.

By iteratively updating the parameters in the opposite direction of the gradient, the algorithm gradually moves closer to the minimum until convergence. The learning rate is a hyperparameter that determines the size of the steps taken in the parameter space during each iteration of the algorithm. It controls the speed at which the algorithm converges to the minimum. If the learning rate is too high, the algorithm may overshoot the minimum, bouncing back and forth across it or even diverging altogether. This can prevent the algorithm from converging and lead to unstable and inaccurate results. On the other hand, if the learning rate is too low, the algorithm will take very small steps and converge very slowly. This can result in long training times and a delayed convergence to the optimal solution. Finding an appropriate learning rate is crucial for the gradient descent algorithm to work effectively.

**Step 6 - Repeat:** Steps 2-5 are iteratively repeated for multiple epochs until the network's performance converges or reaches a satisfactory level [67].

## 3.2 Convolutional Neural Network

### 3.2.1 Overview

A CNN is a type of deep learning model that is primarily designed for processing and analyzing structured grid-like data, such as images or time series. It is widely used in computer vision tasks such as image classification, object detection, and image segmentation. The foundational work in CNNs can be attributed to LeCun and his colleagues, who developed the LeNet-5 architecture in 1998. LeNet-5 was primarily designed for handwritten digit recognition and served as a breakthrough in the field of deep learning.

They excel at automatically learning and extracting meaningful features from raw data without manual feature engineering. While modern CNNs are quite complex, they contain three main components: convolutional layers, pooling layers, and fully connected layers. Overall, CNNs are essential in computer vision and deep learning due to their flexibility, effectiveness, and capacity to learn hierarchical representations from visual data. Fig. 3.4 presents a visual example of a CNN [3].

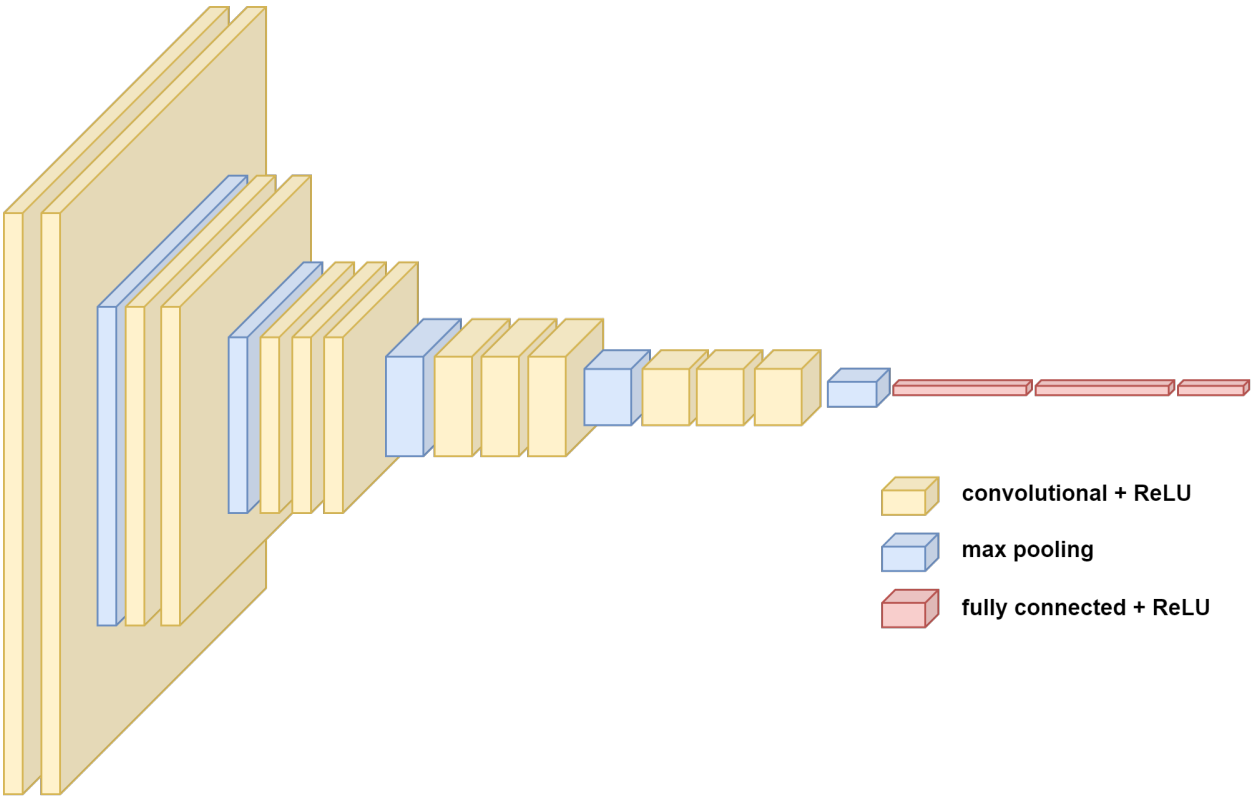



**Figure 3.4:** An illustration of a 2-D CNN employed for image classification. In this example, the CNN comprises five sets of convolutional and max pooling layers, followed by three fully connected layers.

### 3.2.2 Convolutional Layers

Convolutional layers are an essential component of CNNs and are primarily designed to extract meaningful features from input data. Convolutional layers are inspired by the concept of convolution from mathematics and signal processing. The convolutional layer operates on a multi-dimensional input, typically a 2D image or a 3D volume, and applies a set of learnable filters to the input. These filters, also known as convolutional kernels or feature detectors, are small-sized matrices with learnable weights. The convolution operation described by (3.8) involves sliding a kernel K with dimensions $P \times Q$ across the input signal I with dimensions $M \times N$. In this context, $Y(i,j)$ denotes the value at position $(i,j)$ in the output feature map. The double summation involves iterating over all possible values of $m$ and $n$ in the kernel. For each relative position $(m,n)$, the input signal value $I(i+m, j+n)$ is multiplied with the kernel value $K(m,n)$ and these

products are accumulated over all possible relative positions. The final result of this accumulation is the value at position $(i, j)$ in the output signal $Y$. The resulting feature map dimensions are $(P - M + 1) \times (Q - N + 1)$.

$$Y(i,j) = (I * K)(i,j) = \sum_{m=0}^{P-1} \sum_{n=0}^{Q-1} I(i+m, j+n) K(m,n). \tag{3.8}$$

Each element in the feature map captures a local pattern or characteristic that the kernel is designed to detect. The early convolutional layers in a CNN capture low-level features like edges and textures. As the information flows through the network, subsequent layers capture more sophisticated features like shapes and objects. An illustration of the convolution operation is shown in Fig. 3.5.

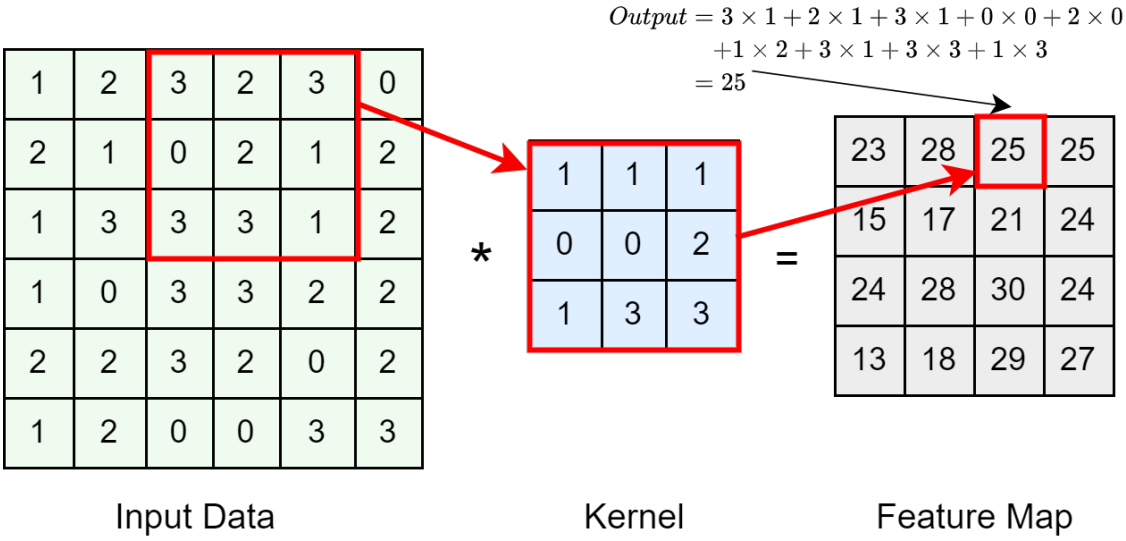



**Figure 3.5:** Illustration of a convolution operation. In this depiction, a kernel traverses the input data and computes dot products to generate a feature map. The red rectangular boxes highlight the computations performed for one such element in the feature map.

After convolution, a non-linear activation function, such as ReLU, is often applied element-wise to introduce non-linearity and capture complex relationships between features. The activation function helps in enhancing the expressive power of the model. Convolutional layers commonly incorporate a bias term added to the output of each filter. This bias term allows the model to learn an additional offset, providing greater flexibility in representing features [3, 67].

### 3.2.3 Pooling Layers

Pooling layers play a crucial role in downsampling the spatial dimensions of feature maps, which refers to reducing the spatial dimensions of input data while preserving information. Pooling is often done to make computation more efficient and manageable. Pooling layers operate on each feature map independently and divide them into regions that can be non-overlapping or overlapping. Each pooling layer uses a pooling window that slides over the input feature map to define regions to be downsampled. The pooling operation then aggregates the information within the pooling window to produce a single output value. Stride refers to the step size used to move the pooling window across a feature map. Overall, pooling layers help the network become less sensitive to small spatial variations in the input, which is known as translational invariance. Average pooling and max pooling are two commonly used types of pooling layers.

**Average pooling:** Average pooling computes the average value of a region within the input feature map. Average pooling provides a smoothing effect and can help reduce variations and noise in the feature map. The mathematical formulation for average pooling is provided by (3.9), where $F_{in}$ denotes the input feature map, $R \times S$ represents the dimensions of the pooling window, and vertical and horizontal strides are indicated by $P$ and $Q$, respectively. By summing the values within the pooling window and subsequently averaging them over the total number of elements in the window, $R \times S$, the resulting average value for position $(i, j)$ in the output feature map $F_{out}$ is obtained. A visual representation of the average pooling process is depicted in Fig. 3.6.

$$F_{out}(i,j) = \frac{1}{R \times S} \sum_{r=1}^{R} \sum_{s=1}^{S} F_{in}((i-1) \times P + r, (j-1) \times Q + s). \tag{3.9}$$

**Max pooling:** Max pooling takes the maximum value within each pooling region and discards the rest of the information. It also divides the feature map into non-overlapping regions and selects the maximum value from each region. As a result, it tends to create sparse feature representations that capture the most important features in a feature map. The operation is defined by (3.10), where $F_{in}$ represents the input feature map, $R \times S$ denotes the dimensions of the pooling window, and $P$ and $Q$ represent the vertical and horizontal strides respectively. The max pooling operation

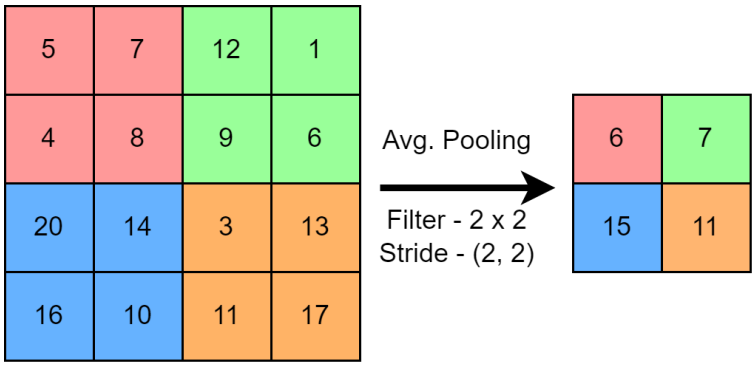



**Figure 3.6:** An illustration of average pooling.

aggregates the maximum value from the pooling window's elements to produce the resulting maximum value for position $(i, j)$ in the output feature map $F_{out}$. An illustrative representation of max pooling is provided in Fig. 3.7.

$$F_{\text{out}}(i,j) = \max\left\{F_{\text{in}}((i-1)\times P + r, (j-1)\times Q + s) \mid 1 \le r \le R, 1 \le s \le S\right\}. \tag{3.10}$$

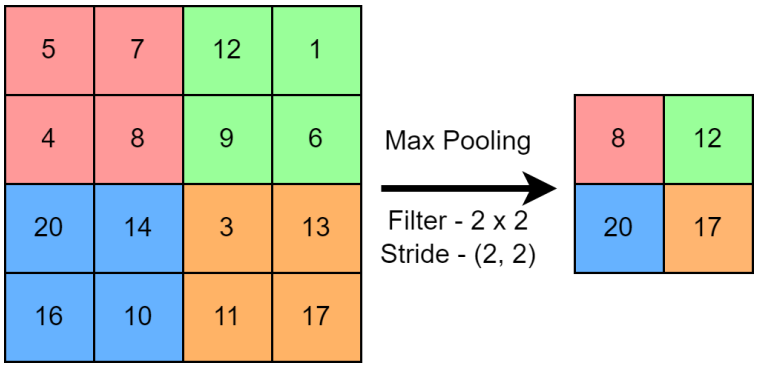



**Figure 3.7:** An illustration of maximum pooling.

### 3.2.4 Fully Connected Layers

FC layers, also referred to as dense layers, transform the output of previous layers into the final output predictions. These layers are designed to capture complex relationships between features and enable the network to learn high-level abstractions from the input data. Fully connected layers aid in generalization by capturing abstract representations, allowing the network to recognize similar patterns across different contexts. This improves its ability to generate predictions for unseen

data. FC layers exhibit an identical structure and perform the same computations as those found in an MLP network. In an FC layer, each neuron is connected to every neuron in the previous layer, forming a fully connected graph. This means that the outputs of all neurons in the previous layer serve as inputs to each neuron in the fully connected layer. The connections are represented by weights, which determine the strength and impact of the input signals on the neuron's activation [67].

## 3.3 LSTM

### 3.3.1 Overview

A traditional LSTM network is a type of RNN architecture that is widely used in the field of deep learning for sequence modeling and time series analysis. It was introduced by Hochreiter and Schmidhuber in 1997 to mitigate the vanishing gradient problem that affects traditional RNNs. The vanishing gradient problem refers to the issue where gradients during backpropagation diminish exponentially as they propagate backward through layers, which may cause stalled learning in deep neural networks. The key idea behind an LSTM is to introduce a memory cell that can store information over long periods of time, which allows the network to capture and learn dependencies in sequences more effectively. The memory cell is responsible for remembering or forgetting information based on its relevance and importance. By selectively updating and forgetting information through the gates, the LSTM can learn long-term dependencies in sequences and make predictions based on the relevant context. This makes it particularly effective for tasks such as speech recognition, language translation, sentiment analysis, and time series forecasting.

### 3.3.2 Traditional LSTM

The LSTM architecture effectively processes sequential data by using multiple interconnected components. As illustrated in Fig. 3.8, LSTMs employ three gating functions: the input gate,

forget gate, and output gate. These gates control the flow of information into and out of the cell state, allowing the network to decide what to remember and what to discard.

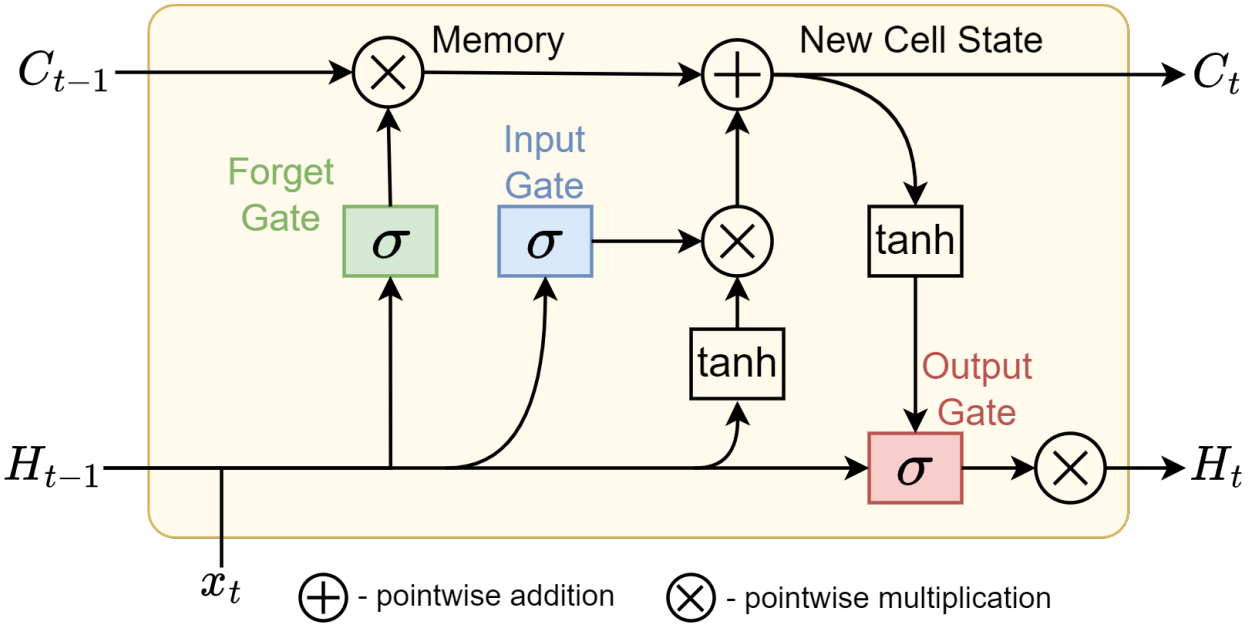



**Figure 3.8:** A visual representation illustrating a typical LSTM cell that contains three gates responsible for regulating the flow of information. In this depiction, $\mathbf{x}_t$, $\mathbf{C}_t$, and $\mathbf{H}_t$ represent the input data from a time series, the cell state, and the hidden state respectively, at a specific timestamp $t$.

The input gate $i_t$ determines the relevance of the current input $x_t$ and the previous hidden state $H_{t-1}$ in updating the cell state at a specific time step $t$. The input gate performs linear transformations on both the current input and the previous hidden state. This entails using two distinct weight matrices: $W_{xi}$ is responsible for the current input, and $W_{hi}$ is responsible for the hidden state from the previous time step. The $*$ symbol denotes the Hadamard product operation which facilitates element-wise multiplication between two matrices of the same dimensions. Additionally, a bias term $b_i$ represents a scalar value that gets added to the weighted summation of the input data and the previous hidden state. The sigmoid activation function defined in (3.2) is used to calculate the extent to which different components of the input should influence the cell state. The gate output is a value between 0 and 1, where 0 would indicate that the output has no influence on the cell state

update, and 1 would indicate full influence. The equation representing the input gate is defined in (3.11).

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * H_{t-1} + b_i). \tag{3.11}$$

Conversely, the forget gate $f_t$ decides which information from the previous cell state $C_{t-1}$ should be retained and what should be discarded. The forget gate performs linear transformations on both the current input and the previous hidden state. To accomplish this, two weight matrices are employed: $W_{xf}$ manages the impact of the current input, and $W_{hf}$ manages the influence of the hidden state from the previous time step. Additionally, a bias term $b_f$ is added to the weighted summation of the input data and the previous hidden state. It employs the sigmoid function to generate a forget vector, determining the degree of information to be discarded. The equation describing the forget gate is defined in (3.12).

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * H_{t-1} + b_f), \tag{3.12}$$

The output gate $o_t$ contributes to the capability of the LSTM cell to selectively expose or hide information from the current cell state $C_t$ and the hidden state $H_t$. The output gate also performs linear transformations on both the current input and the previous hidden state. This is achieved by using two weight matrices: $W_{xo}$ determines the impact of the current input, and $W_{ho}$ determines the impact of the hidden state from the previous time step. Additionally, a bias term $b_o$ is added to the weighted summation of the input data and the previous hidden state. To quantitatively determine the extent of information to be transmitted from the present time step to the final hidden and cell states, the output gate leverages the sigmoid function. The equation describing the output gate is defined in (3.13).

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * H_{t-1} + b_o), \tag{3.13}$$

The cell state $C_t$ serves as a memory unit that retains information over long sequences. Its computation is governed by the expression given in (3.14). A pointwise multiplication operation

is performed between the forget gate $f_t$ and the previous cell state $C_{t-1}$. This operation modulates how much of the previous cell state should be carried forward to the current time step.

Similarly, a pointwise multiplication is performed between the input gate $i_t$ and the candidate values. The candidate values are computed by combining the current input $x_t$ with the previous hidden state $H_{t-1}$, both of which are transformed using weight matrices $W_{xc}$ and $W_{hc}$, along with a bias term $b_c$. To ensure appropriate scaling and regulation, a $tanh$ operation is applied to these candidate values. The scaled candidate values undergo an element-wise multiplication with the input gate $i_t$. The outcomes from both element-wise multiplication processes are amalgamated using an element-wise addition operation, resulting in the generation of the current cell state $C_t$.

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * x_t + W_{hc} * H_{t-1} + b_c), \tag{3.14}$$

The hidden state $H_t$ carries information about the network's understanding of the sequence up to the current time step, as computed in (3.15). The new hidden state $H_t$ is computed using an element-wise multiplication between $o_t$ and $tanh(C_t)$.

$$H_t = o_t \circ \tanh(C_t). \tag{3.15}$$

### 3.3.3 Bidirectional LSTM

In a standard LSTM, the information flows from the past to the future. However, in many tasks, such as speech recognition and language translation, the context of a particular sequence element depends not only on the preceding elements but also on the subsequent elements. Bidirectional LSTMs address this limitation by running two separate LSTM networks simultaneously, one processing the sequence in the forward direction and the other in the backward direction, as depicted in Fig. 3.9. The forward LSTM computes hidden states $h_t^{(f)}$ for each time step $t$ from 1 to $T$, where $T$ represents the number of hidden cells. On the other hand, the backward LSTM computes hidden states $h_t^{(b)}$ from $T$ to 1. The final bidirectional LSTM hidden state $h_t^{(bi)}$ at each time step is obtained by concatenating the forward and backward hidden states [67].

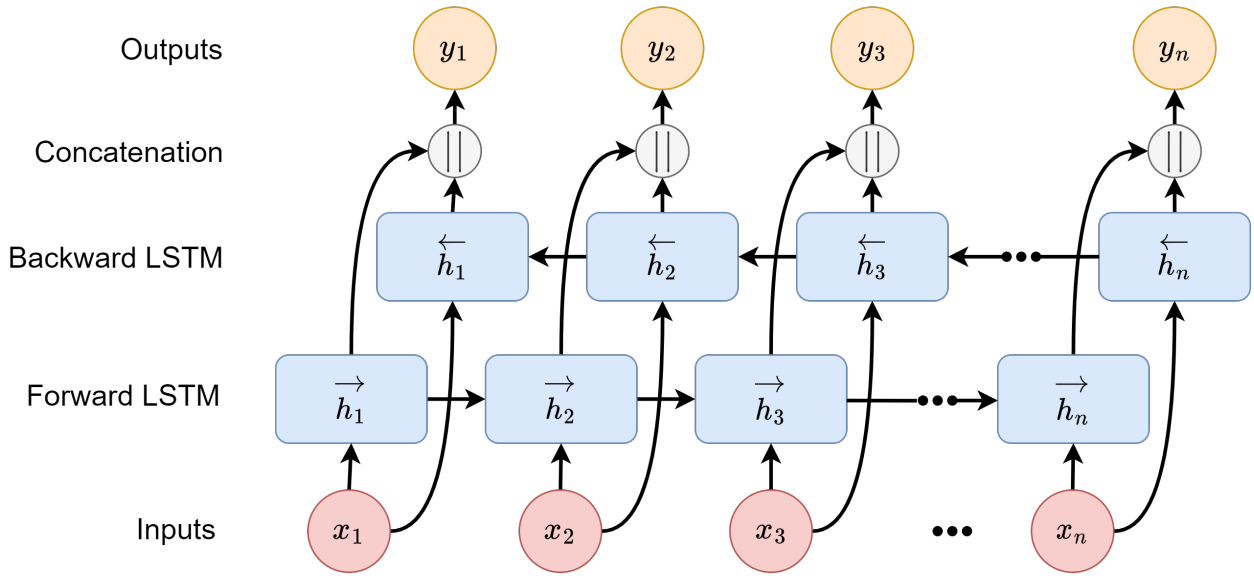



**Figure 3.9:** A general illustration of a Bi-LSTM. In this depiction, $\overrightarrow{h}$ represents a hidden state in the Forward LSTM network whereas $\overleftarrow{h}$ represents a hidden state in the Backward LSTM network.

## 3.4 Attention

### 3.4.1 Overview

Attention has revolutionized the field of DL. Attention allows models to focus on relevant parts of the input, selectively attending to the most informative features or regions. This ability to prioritize relevant information is crucial when dealing with complex data, as it enables models to extract salient patterns and discard irrelevant or noisy elements. By assigning higher weights to important components, attention mechanisms effectively enhance the model's discriminative power, leading to improved performance on various tasks. In addition, attention mechanisms facilitate the handling of long-range dependencies in sequential or spatially distributed data. Traditional RNNs may struggle with capturing dependencies that span long distances, leading to information loss or limited modeling capabilities. Attention provides a solution by allowing the model to dynamically attend to different parts of the input, regardless of their temporal or spatial separation. This enables the model to effectively capture dependencies across long sequences or capture the global context in spatially distributed data.

Attention mechanisms can be incorporated into various DL architectures, including recurrent RNNs, CNNs, and transformer models. Attention has played a pivotal role in achieving state-of-the-art performance in the domains of computer vision and NLP. For example, attention mechanisms have been successfully employed in NLP for tasks in language translation [68], sentimental analysis [69], and text summarization [70]. This flexibility allows attention to be seamlessly integrated into existing architectures or customized to suit specific task requirements [67].

### 3.4.2 Self-Attention

Self-attention, also known as scaled dot-product attention, allows a model to focus on different parts of the input sequence to capture both local and global dependencies. The input sequence is typically represented as a set of vectors, where each vector corresponds to an element in the sequence. The sequence of input vectors are denoted by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where $\mathbf{x}_i$ represents the $i$-th input vector.

The process of self-attention involves computing three key components: query, key, and value. The query, key, and value matrices are denoted as $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, respectively. These components are learned during the training process and are linear projections of the input vectors, as mathematically expressed in (3.16) to (3.18).

$$\mathbf{Q} = \mathbf{W}_Q \mathbf{X}, \tag{3.16}$$

$$\mathbf{K} = \mathbf{W}_K \mathbf{X}, \tag{3.17}$$

$$\mathbf{V} = \mathbf{W}_V \mathbf{X}, \tag{3.18}$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learnable weight matrices that project the input vectors into the query, key, and value spaces, respectively.

The attention score between the $i$-th position and the $j$-th position in the sequence is obtained by taking the dot product between the query vector of the $i$-th position and the key vector of the $j$-th position. The division by $\sqrt{d_k}$ is introduced to prevent the attention scores from becoming too large, which could lead to instability during the training process. The attention scores represent

the relevance or similarity between different positions in the sequence. To obtain the attention weights, which indicate the importance of each position with respect to the current position, the softmax function described in Section 3.1.3 is applied to the attention scores. The softmax function normalizes the attention scores which produces a probability distribution that sums to 1.

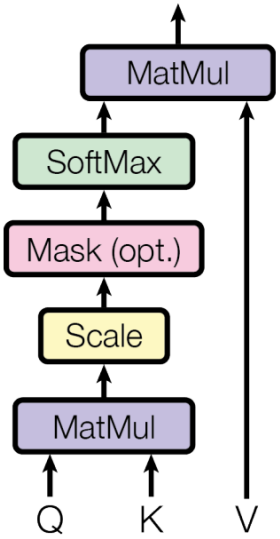



**Figure 3.10:** A flow diagram illustrating the operations performed for computing self-attention [4].

Using the attention weights, a weighted sum of the value vectors is computed to obtain the output representation at each position. The value vectors capture the information associated with each position in the input sequence. The weighted sum combines the values from different positions based on their corresponding attention weights, allowing the model to focus more on important positions and less on irrelevant ones. Fig. 3.10 presents a flow diagram showing the computational steps needed to perform self-attention. Mathematically, computing the self-attention for a sequence of input vectors is expressed in (3.19).

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{\mathbf{Q}_i \cdot \mathbf{K}_j}{\sqrt{d_k}})V. \tag{3.19}$$

### 3.4.3 Multi-head Attention

Multi-head attention is an extension of the self-attention mechanism that enhances the ability of self-attention to capture dependencies within a sequence but incorporating multiple attention

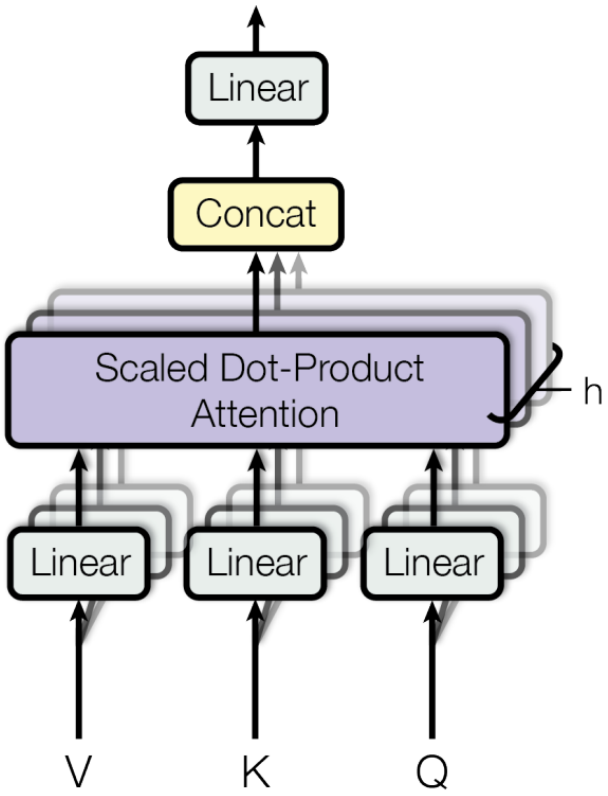



**Figure 3.11:** A flow diagram illustrating the operations performed for computing multi-head attention [4].

heads. Each attention head can focus on different parts of the input sequence, attending to different patterns or relationships. By allowing the model to jointly attend to different aspects of the input, multi-head attention facilitates richer and more expressive representations.

In multi-head attention, a sequence containing $d_{model}$ input vectors is used as input. The self-attention mechanism is applied $h$ times in parallel, each with its own set of the learned query, key, and value weight matrices with $d_k$, $d_k$, and $d_v$ dimensions, respectively. These matrices are usually smaller in dimension compared to single-head attention. Next, each attention head computes its own attention scores and attention weights following the self-attention mechanism, as described previously. The output of each attention head is concatenated together to form a single vector and is passed through a linear projection layer to reduce its dimensionality, as defined in (3.20). This projection helps the model aggregate the information from different attention heads effectively [4, 67].

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, ...\text{head}_2^{\text{out}}, \ldots, \text{head}_h^{\text{out}})\mathbf{W}_O, \tag{3.20}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. The projections denote parameter matrices where $\mathbf{W}_{Q_i} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_{K_i} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_{V_i} \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $\mathbf{W}_O \in \mathbb{R}^{h \times d_v \times d_{\text{model}}}$.

# Chapter 4

# An Efficient CNN-BiLSTM Model for Multi-class Intracranial Hemorrhage Classification

A deep learning solution has been developed with the purpose of automatically identifying ICH. The model uses windowing as a preprocessing step for the input CT scan images. This technique enhances the contrast of each CT scan slice, enabling better detection of subtle abnormalities associated with ICH. By applying an extensive set of data augmentations to the CT scan slices, the aim is to increase the diversity and variability of the training data.

In order to learn distinct feature representations of ICH, a 2-D CNN model is used. Subsequently, a BiLSTM network is utilized to capture sequential patterns among the CT scan slices. The LSTM network takes the feature embeddings generated by the CNN and leverages them to extract temporal dependencies and long-range interactions. This integration of CNN and LSTM allows the model to effectively analyze the sequential nature of CT scan data, leading to improved predictive accuracy.

For evaluation purposes, the trained model generates multi-label predictions for each sample in the RSNA-ICH test set, CQ500, and PhysioNet-ICH datasets. The performance of the model is measured using key metrics, such as sensitivity, specificity, precision, and the AUC score. These

metrics provide a comprehensive assessment of the model's ability to correctly identify instances of ICH.

## 4.1 Proposed Solution

An elaborate functional flow diagram depicting the CNN-BiLSTM CT scan classification framework is presented in Fig. 4.1. The framework encompasses three key phases: **Phase 1** involves data preprocessing, **Phase 2** focuses on model development and training, and **Phase 3** includes model testing and evaluation.

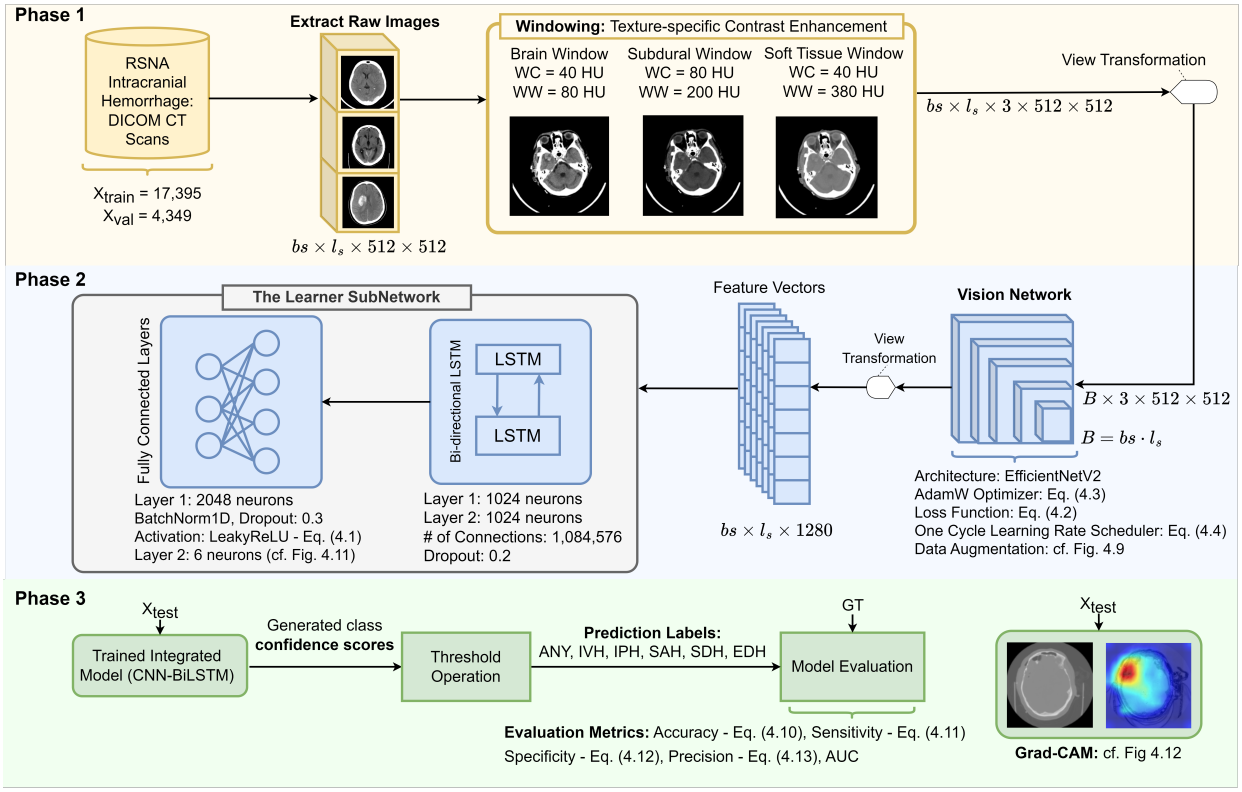



**Figure 4.1:** Detailed functional flow diagram of the proposed CNN-BiLSTM CT scan classification framework. It consists of three abstract phases – **Phase 1**: Data preprocessing, **Phase 2**: Model development and training, and **Phase 3**: Model testing and evaluation.

In Phase 1, extensive data preprocessing is performed to enhance the model's ability to capture relevant features and improve its generalization capabilities. The RSNA 2019 Brain CT Hemorrhage Challenge dataset is partitioned into a training set and a test set. Drawing inspiration from

established radiology workflows, one of the key image preprocessing steps we apply is known as windowing. This technique aims to enhance the contrast of each CT scan slice and highlights subtle differences between healthy and abnormal tissues. More specifically, three commonly used window settings are used: brain, subdural, and soft tissue. Each window configuration emphasizes specific features and improves the visibility of specific anatomical structures and pathological features. Data augmentations are then introduced to further enhance the robustness and diversity of the training.

In Phase 2, the proposed model learns from the preprocessed data to extract meaningful patterns and optimize its parameters to make accurate ICH predictions. Initially, a Vision Network is employed that comprises a 2-D CNN to effectively capture spatial information and local patterns. The Bi-LSTM network takes advantage of the feature embeddings produced by the Vision Network to learn slice-temporal dependencies and long-range interactions. The training process involves iterative epochs, where the model adjusts its parameters to become more accurate at detecting and classifying ICH.

In Phase 3, testing and evaluation are performed to assess the performance, reliability, and generalization capabilities of the proposed model. Initially, the trained model generates multi-label predictions for each sample in the RSNA-ICH test set, CQ500, and PhysioNet-ICH datasets. The performance of the model is quantitatively measured using metrics of sensitivity, specificity, precision, and AUC. These metrics offer a thorough evaluation of the model's capacity to accurately detect different subtypes of ICH. Grad-cam visualizations provide a qualitative assessment by highlighting the regions in CT scan slices that influenced the model's decision. This may assist radiologists in better understanding the predictions of the proposed solution.

### 4.1.1 Data Preprocessing

**Dataset Curation–** In this study, three publicly available benchmark datasets are used: RSNA, CQ500, and PhysioNet. The RSNA dataset was used for both training and testing while the CQ500 and PhysioNet datasets were used for independent testing. A detailed description of each dataset

is provided below. The scan and slice subtype distribution is tabulated in Table 4.1 and visualized in Fig. 4.4 to Fig. 4.6

This study uses a large collection of non-contrast head CT scans collected by the RSNA association for model training, validation, and testing. Each CT scan slice has a resolution of $512 \times 512$ pixels. The dataset consists of over 25,000 scans, with each scan containing 20 to 60 slices, resulting in a total of 874,034 slices. The samples were annotated by over sixty experienced neuroradiologists to produce ground truth labels. More specifically, each slice was assigned a series of binary labels for each of the six classes: EDH, SDH, SAH, IPH, IVH, or ANY [2]. In this case, the ANY class has a positive prediction label if any of the other five classes have a positive prediction label. For example, the slice *ID_d81ea8751* has a multi-hot[1] encoded target label of $[1, 0, 0, 0, 0, 1]$ indicating that IPH and ANY are present, respectively. Moreover, a slice can contain multiple different ICH subtypes. For example, the slice *ID_6be2c702e* has a multi-hot encoded target label of $[1, 0, 1, 0, 1, 1]$ indicating that IPH, SAH, SDH, and ANY are present, respectively. The dataset is heavily imbalanced, with the majority of samples (86%) belonging to the non-ICH class, while only a small percentage of samples (14%) are part of ICH.

The RSNA dataset was specifically selected for training the model because it offers the largest and most diverse set of CT scans among the three benchmark datasets. This selection ensures a wealth of rich and comprehensive data, facilitating the robust training of our model. The training dataset contains 21,744 CT scans and the test set contains 3,518 CT scans. The training dataset is further partitioned into two subsets: a training set, which accounts for 80% of the CT scans, and a validation set, containing the remaining 20%. This division results in 17,395 CT scans allocated to the training set and 4,349 CT scans allocated to the validation set. The training set is used to train the model's parameters, while the validation set helps to assess the training progression by providing an independent dataset for measuring performance. This separation aids in preventing overfitting and ensures the model's effectiveness on unseen data. This work removes low-quality/empty CT scan slices from the training set when all pixel values are less than $-150\ HU$ before being used as input into the proposed model.

[1]It encodes each type in the input slice into a single array of size ICH types, containing a 1 for each type presents in the sample.

The CQ500 dataset consists of a total of 491 CT scans, including 205 with ICHs, 40 fractures, 65 middle shifts, 127 mass effect, and 54 normal controls. The 205 ICH scans contain all five subtypes, including 28 IVHs, 134 IPHs, 60 SAHs, 13 EDHs, and 53 SDHs. The involvement of multiple radiology centers in New Delhi, India, further enhances the dataset's diversity and representation of cases from different clinical settings. The annotation of each ICH subtype was manually performed by three senior radiologists [19].

The PhysioNet-ICH dataset consists of 75 patients, including 36 with ICH and 39 without ICH, comprising a total of 2,814 slices. The dataset source is from the Al-Hilla Teaching Hospital in Iraq and obtained using a Siemens/SOMATOM Definition AS CT Scanner with a 5 mm slice thickness. The average age of the patients in the dataset is $27.8 \pm 19.5$. Two radiologists conducted the annotation of non-contrast CT scans and classified the ICH subtypes through simultaneous review and consensus on the diagnosis [44].

**Exploratory Data Analysis–**

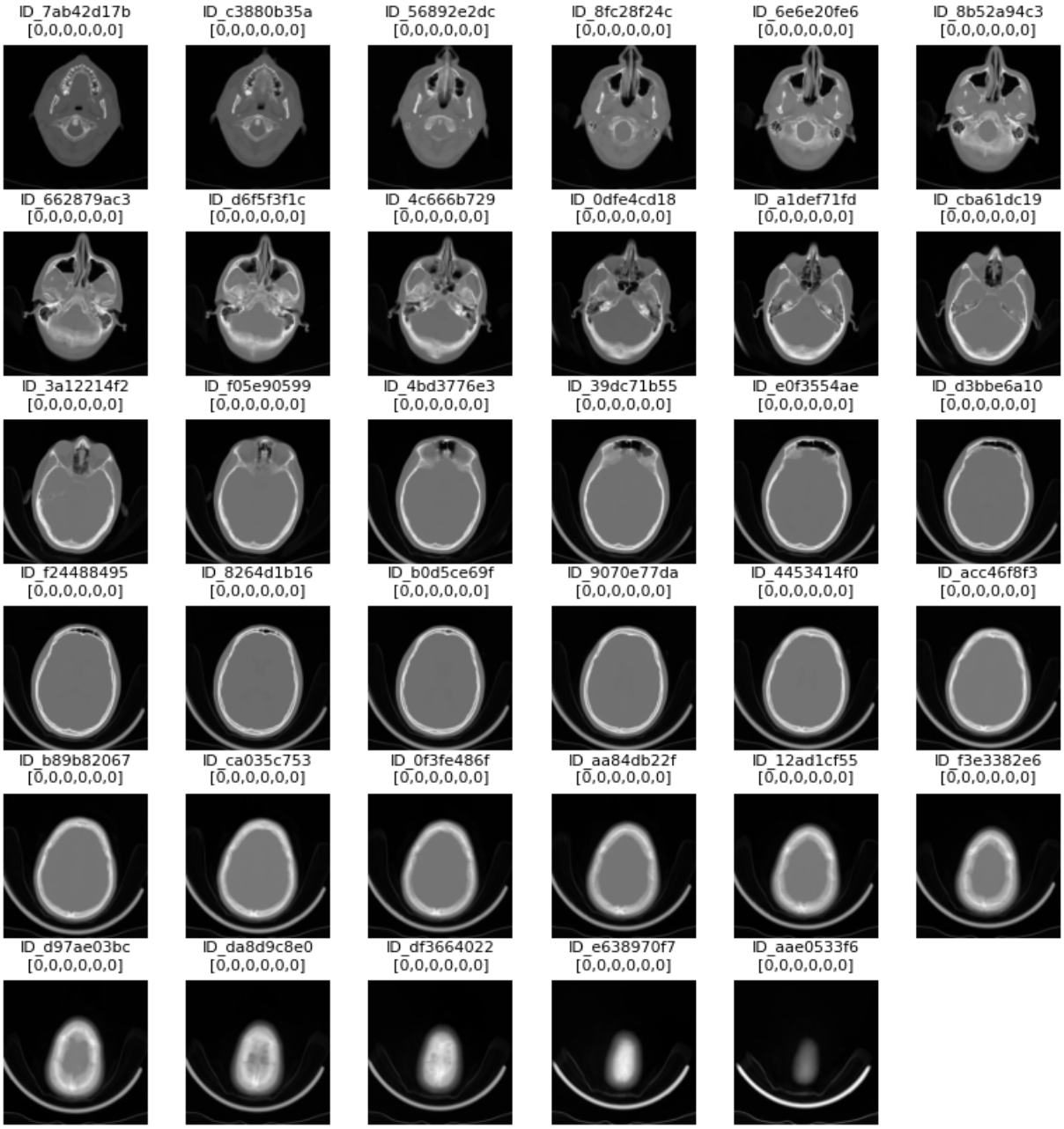



**Figure 4.2:** A visualization of all 35 CT scan slices for the CT scan, ID_ec0310f506, from the RSNA dataset. The CT scan does not contain any slices with ICH. The corresponding image ID and multi-hot target labels are provided above each CT scan slice. The multi-hot target labels are denoted in the following order: [EDH, IPH, IVH, SAH, SDH, ANY].

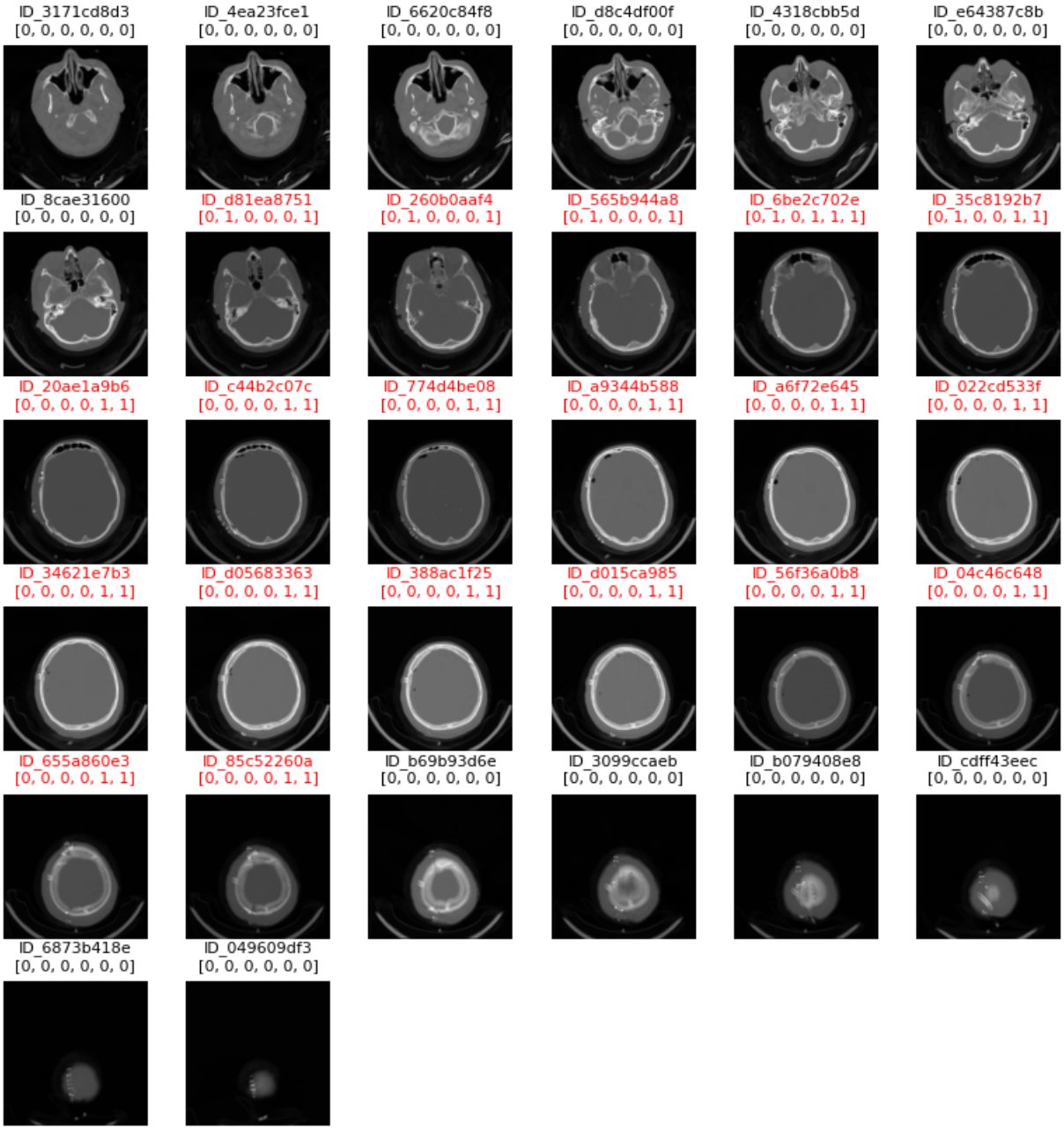



**Figure 4.3:** A visualization of all 35 CT scan slices for the CT scan, ID_ec0310f506, from the RSNA dataset. The corresponding image ID and multi-hot target labels are provided above each CT scan slice. The multi-hot target labels are denoted in the following order: [EDH, IPH, IVH, SAH, SDH, ANY]. CT scan slices with ICH are highlighted with red labels.

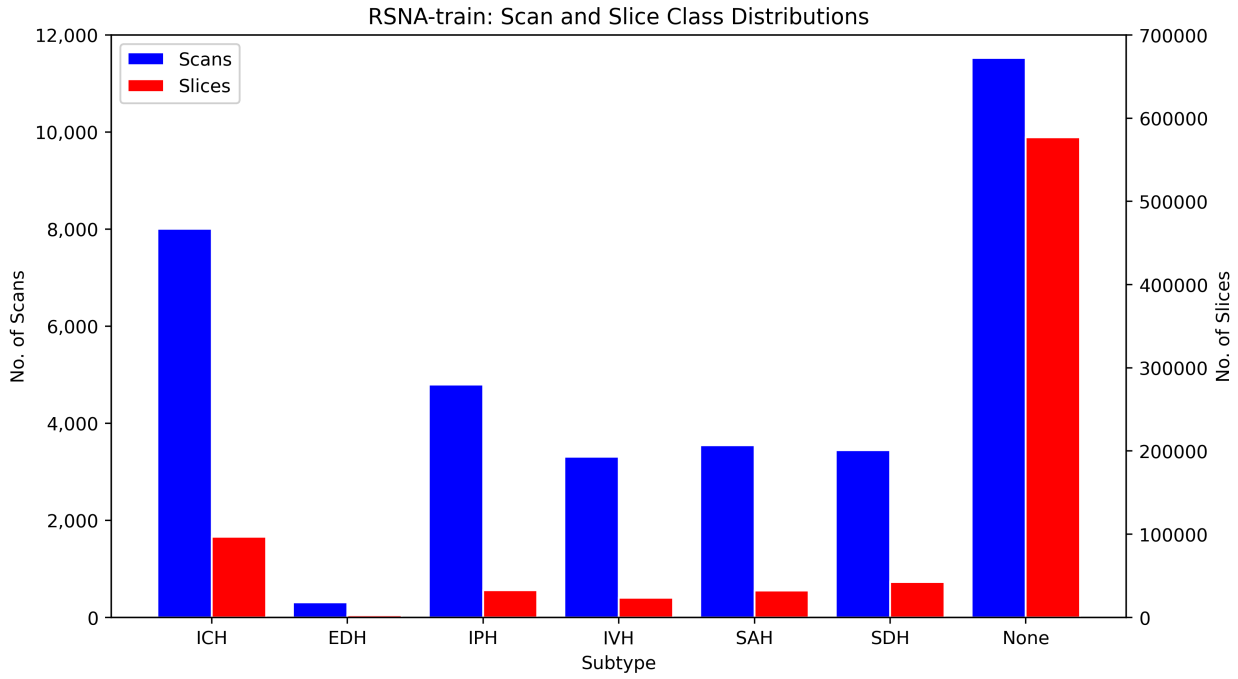



**Figure 4.4:** The scan and slice distribution for the RSNA-train set. Note that ground truth labels have not been provided for the test set.

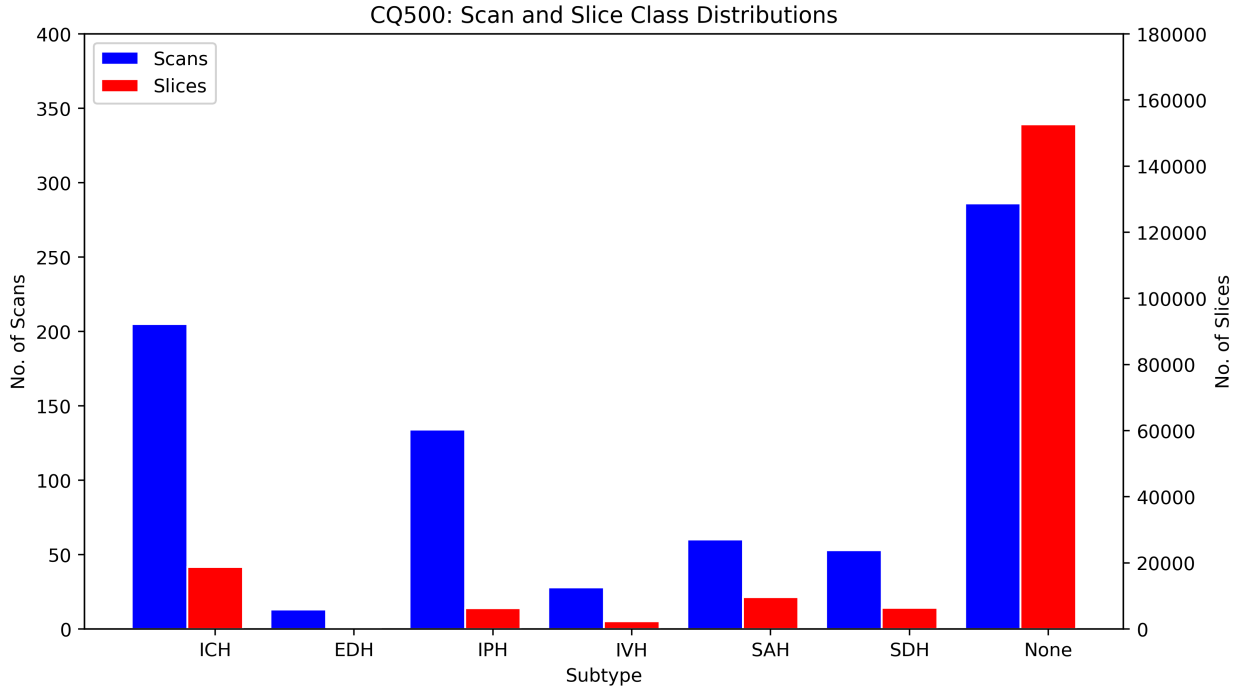



**Figure 4.5:** The scan and slice distribution for the CQ500 dataset.

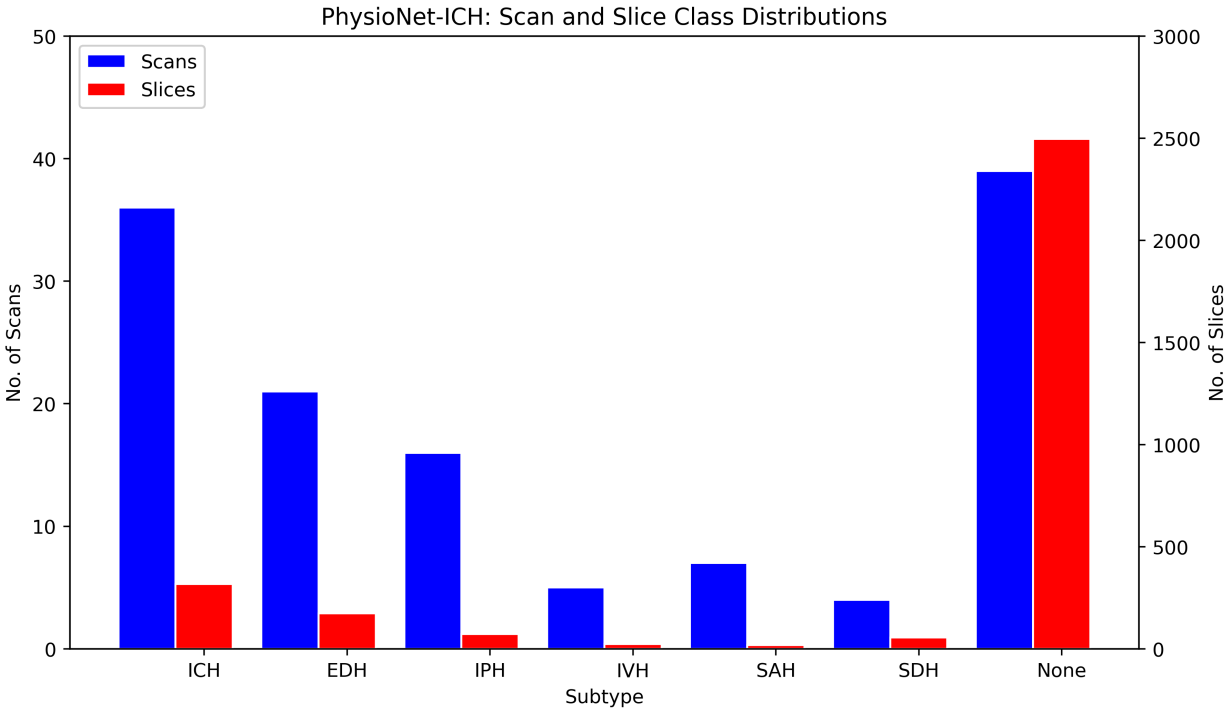



**Figure 4.6:** The scan and slice distribution for the PhysioNet-ICH dataset.

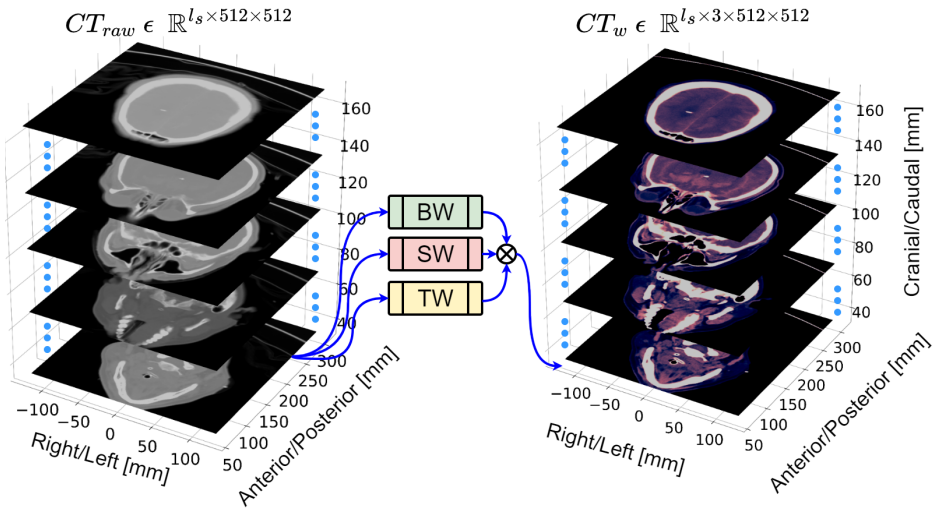
**Table 4.1:** Data distribution characteristics of the utilized benchmark datasets.

| | **RSNA-train** | | **PhysioNet-ICH** | | **CQ500** | |
|---|---|---|---|---|---|---|
| | Scans | Slices | Scans | Slices | Scans | Slices |
| ICH | 8,882 | 107,933 | 205 | 18,774 | 36 | 318 |
| EDH | 354 | 3,145 | 13 | 131 | 21 | 173 |
| IPH | 5,321 | 36,118 | 134 | 6,323 | 16 | 73 |
| IVH | 3,692 | 26,205 | 28 | 2,348 | 5 | 24 |
| SAH | 3,932 | 35,675 | 60 | 9,590 | 7 | 18 |
| SDH | 3,814 | 47,166 | 53 | 6,391 | 4 | 56 |
| None | 12,862 | 644,870 | 286 | 152,616 | 39 | 2,496 |
| Total | 21,744 | 752,803 | 491 | 171,390 | 75 | 2,814 |

**Contrast Enhancement via Windowing–** Each pixel in a CT scan represents a specific value that corresponds to the density of the tissue being scanned. In the case of CT scans, this value is measured in HU. To represent the range of densities, each pixel is assigned a 16-bit value, allowing for 65,536 different grayscale values. However, due to the limitations of the human eye in perceiving subtle differences in grayscale, radiologists employ a technique called windowing to enhance the contrast of CT scan slices. Windowing involves mapping the grayscale values to a different range that highlights specific structures or abnormalities, making them easier to detect. A window is defined by two parameters: the window center (WC) and the window width (WW) [25]. The WC determines the midpoint of the grayscale range that will be mapped, while the WW defines the width of the range. By adjusting these parameters, radiologists can create different window configurations to help visualize specific anatomical regions or pathologies.

In this work, three commonly used window settings are employed to mimic the radiology workflow: the brain window (WC = 40, WW = 80), the subdural window (WC = 80, WW = 200), and the soft tissue window (WC = 40, WW = 380). This creates a three-channel contrast-enhanced image that will be the input to the feature extractor of the proposed model (cf. Section 4.1.2 - The Vision Subnetwork).

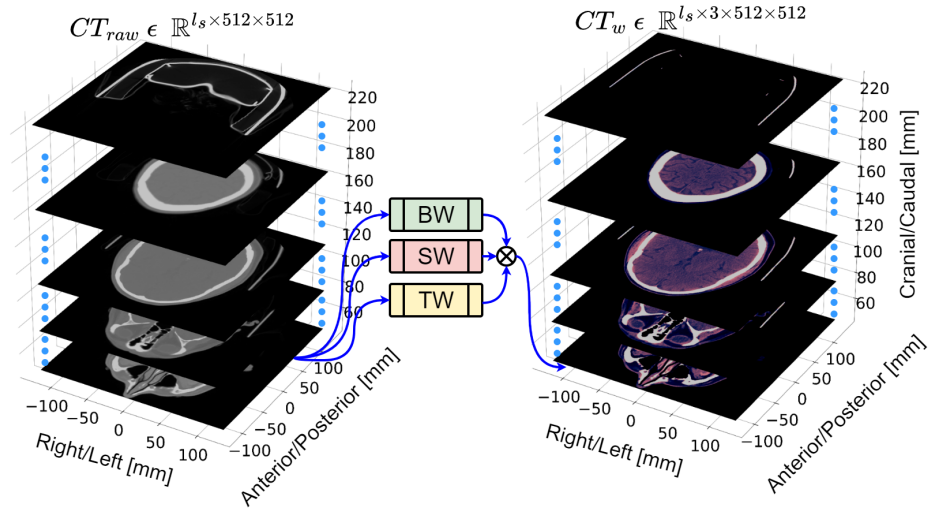
Fig. 4.7 and Fig. 4.8 illustrate the windowing process used in this work. These figures demonstrate the application of windowing to an actual CT scan sample, one showing an ICH and the other without ICH.




a) A raw head CT scan with ICH

b) Input after windowing

**Figure 4.7:** An example of the windowing process using an ICH CT scan, SeriesInstanceUID: ID_4ac84839aa having 28 slices. For visual clarity, five axial slices in positions 1, 8, 15, 21, and 28 are shown. BW, SW, TW, stand for the windowing operation on each slice using windows of the brain, subdural and soft tissue. $l_s$ - total # of slices in the CT Scan, $\bigotimes$ - Concatenation to form a 3-channel representation (like RGB) for each slice.




a) A raw head CT scan without ICH

b) Input after windowing

**Figure 4.8:** An example of the windowing process using a non-ICH CT scan, SeriesInstanceUID: ID_d6ba679446 having 44 slices. For visual clarity, five axial slices in positions 1, 12, 23, 33, and 44 are shown. BW, SW, TW, stand for the windowing operation on each slice using windows of the brain, subdural and soft tissue. $l_s$ - total # of slices in the CT Scan, $\bigotimes$ - Concatenation to form a 3-channel representation (like RGB) for each slice.

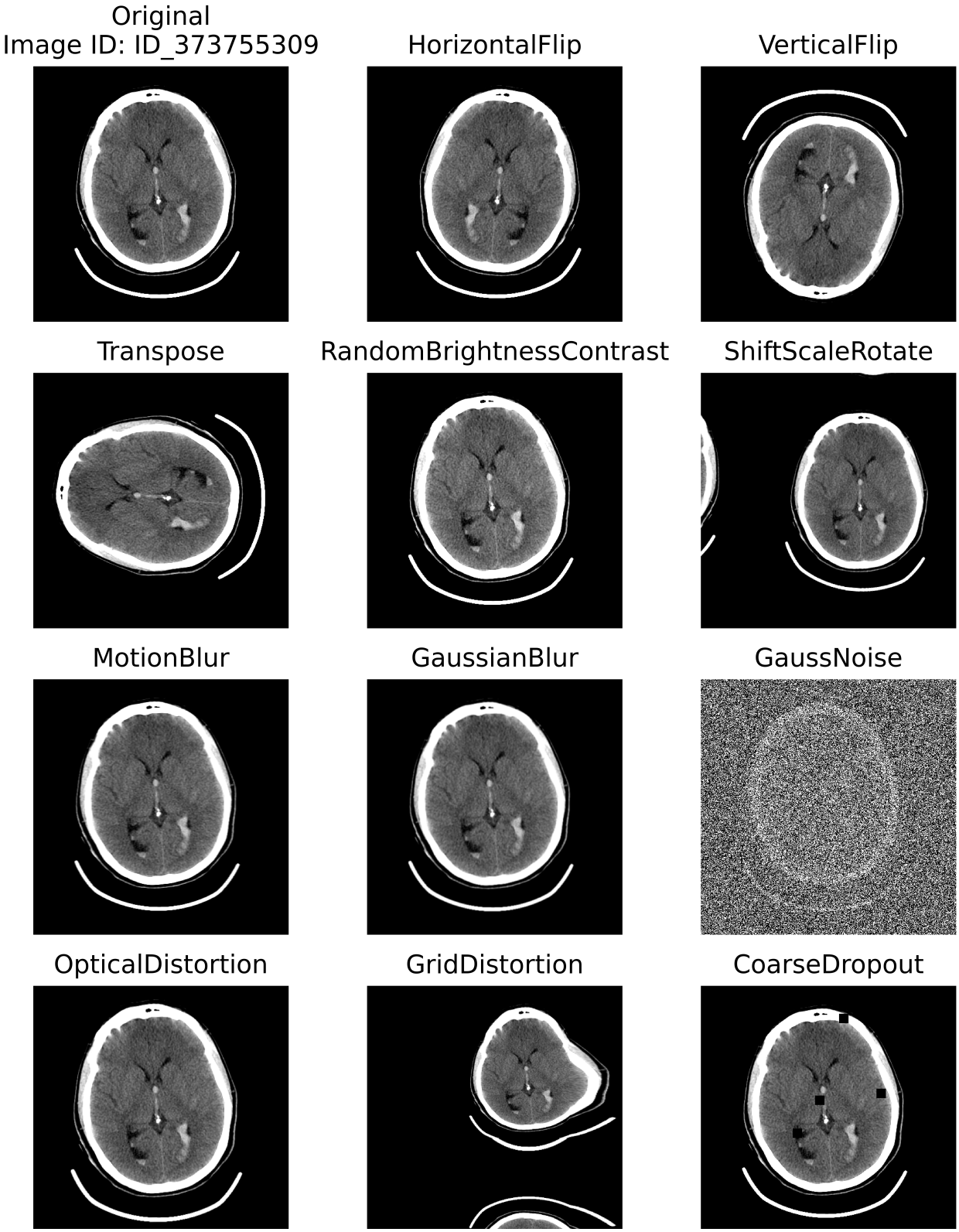



**Figure 4.9:** An example of the application of various data augmentation strategies used in this work.

**Data Augmentation–** This is a data regularization process that refers to a set of techniques that apply geometric image transformations to the original 2D slices of the CT scans. By synthetically creating more diverse training data, data augmentations can help to prevent the model from overfitting as it will allow the model to learn generalized representations of the ICH. In this study, an extensive set of eleven data augmentations are applied: horizontal flipping, vertical flipping, transposing, random brightness and contrast adjustments, shift, scale, rotations, motion blur, median blur, Gaussian blur, Gaussian noise, optical distortion, grid distortion, and coarse dropout as an example shown in Fig. 4.9.

### 4.1.2 The Vision Subnetwork

A pre-trained vision network, EfficientNetV2-Small, serves as a CNN model that learns to extract features after being fine-tuned on the RSNA dataset [71].

EfficientNet is a family of CNN architectures designed to achieve a balance between model size and accuracy, making it suitable for a wide range of applications. The EfficientNet models have achieved state-of-the-art performance on various computer vision tasks while being computationally efficient.

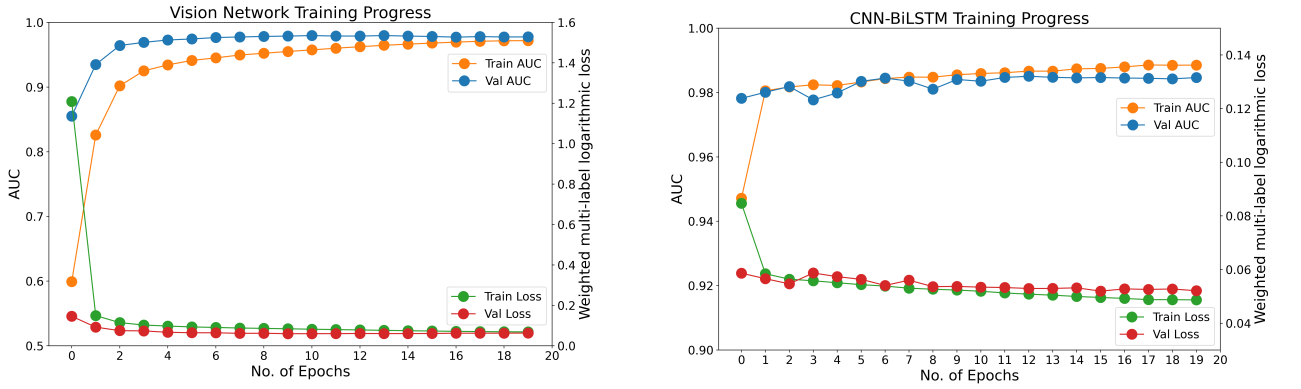
The EfficientNet architecture has been derived based on the MobileNetV2 architecture. The key idea behind EfficientNet is compound scaling, which involves scaling the dimensions of the network in a principled manner. Instead of manually adjusting individual hyperparameters such as depth, width, and resolution, EfficientNet uses a single scaling parameter. This scaling parameter is applied to depth, width, and resolution simultaneously to obtain the optimal network architecture [72].

One noteworthy aspect of the EfficientNetV2 architecture is its default pooling configuration, which employs adaptive average pooling. However, in this case, the pooling configuration was modified to use max pooling instead. Max pooling is particularly well-suited for the task at hand, as it enhances the network's ability to detect abnormalities, specifically ICH in this context [71].

The output of the vision network is a feature map with dimensions of $bs \times l_s \times 1280$. Here, $bs$ represents the batch size of CT scans, $l_s$ denotes the input sequence length of CT scan slices,

and 1280 corresponds to the dimensionality of the feature representation for each CT scan slice. This feature map consists of a sequence of feature vectors, with each vector capturing the salient information extracted from a particular CT scan slice. The sequence of feature vectors obtained from the vision network is then fed as input to the learner subnetwork.

### 4.1.3 Learner Subnetwork




(a) Training progress of the vision network

(b) Training progress of the integrated model–CNN-BiLSTM

**Figure 4.10:** Training progress of the proposed ICH classification model. During the training of the CNN-BiLSTM, the vision network's parameters are not updated, i.e., kept frozen.

The learner subnetwork in the proposed methodology, illustrated in Fig. 4.1 - Phase 2, consists of two main components: the sequence learning module and the classification module. The sequence learning module is designed as a two-layer sequence-to-sequence BiLSTM network. Each layer of the BiLSTM contains 1024 neurons, resulting in a total of 1,084,576 connections. The purpose of this module is to capture and learn the sequential correlations that exist among adjacent axial CT scan slices. By leveraging the features extracted by the vision subnetwork, the sequence learning module can understand the temporal dependencies and patterns present in the input data. To ensure consistency and compatibility with the RSNA dataset, the input sequence length is set to 60, which corresponds to the maximum number of slices in a CT scan. However, if a particular CT scan contains fewer than 60 slices, zero-based padding is applied to the end of the sequence. This padding strategy ensures that the input sequence length remains consistent across all samples. During the validation process, the model predictions and target labels for the padded images are

removed. This prevents the model from becoming biased toward empty images and ensures a fair evaluation of its performance.

The classification module is implemented as a two-layer fully connected (FC) network. This module receives the learned sequential information from the Bi-LSTM and performs further processing to classify the input into specific subtypes of ICH. The FC network projects the sequential information onto a lower-dimensional space and synthesizes it to make accurate subtype predictions. In order to enhance the training process and improve generalization, several techniques are applied within the classification module. Batch normalization is utilized to normalize the output of the FC layers. This normalization step helps stabilize the training process by reducing the internal covariate shift and allowing the network to learn more effectively. To mitigate overfitting, a dropout mechanism is employed. Dropout randomly sets a fraction of input units to zero during training, which helps prevent the network from relying too heavily on specific features and encourages the learning of more robust representations. By reducing overfitting, dropout enables the model to generalize better and perform well on unseen data. Additionally, the Leaky Rectified Linear Unit (Leaky ReLU) activation function defined in (4.1) is employed in the first layer of the FC network. The Leaky ReLU function introduces non-linearity into the classifier, allowing the network to model complex relationships between features. It also helps address the vanishing gradient problem, which can hinder the training process in deep neural networks.

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha \cdot x, & \text{otherwise} \end{cases}, \tag{4.1}$$

where $x$ is the input to the function, and $\alpha$ is a small positive slope for negative input values that helps to prevent dead neurons.

As illustrated in Fig. 4.11, the second FC layer is linear and contains six neurons that produce class confidence scores. This layer does not contain any activation functions as the objective function defined in (4.2) inherently performs a Sigmoid operation that outputs confidence scores between 0 and 1. During inference, a Sigmoid operation is performed after the 2nd FC layer for predicting the multi-hot classification label as shown in Fig. 4.11-(b). The sequence learning and

the classification modules use a dropout rate of 0.2 and 0.3, respectively during training to reach a generalized solution.

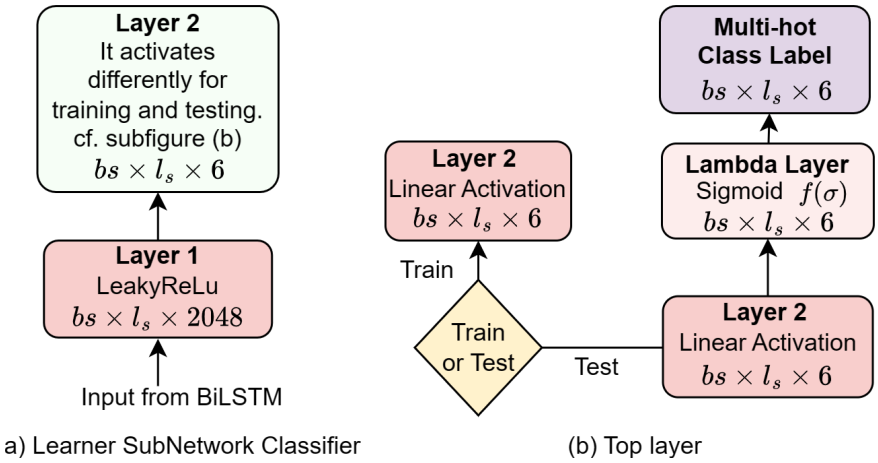



**Figure 4.11:** The top layer of the CNN-BiLSTM. It behaves differently during training and testing.

### 4.1.4 Training Strategy

**Loss function–** During training, the optimizer minimizes the weighted multi-label binary cross entropy with logits loss (BCE_log) defined in (4.2). This loss function also served as the evaluution metric in the RSNA competition. This loss function was used to ensure that the model's training process is directly aligned with the competition's main objective. The loss function minimizes any discrepancy between what the model optimizes for during training and what it's evaluated on during testing. It is widely used in binary classification tasks as it results in stable gradients during optimization. This is especially important when using gradient-based optimization algorithms, like the AdamW optimizer. Stable gradients contribute to faster convergence during training. In addition, weighted multi-label binary cross entropy with logits loss accommodates adjusting class weights. This is valuable when dealing with imbalanced datasets where one class has significantly more samples than the other. By assigning appropriate weights to each class, the loss function can give more importance to underrepresented classes, aiding in better model generalization.

$$\text{BCE\_log} = -\frac{1}{N}\frac{1}{C}\sum_{i=1}^{N}\sum_{j=1}^{C}(w_j y_{i,j}\log(\hat{y}_{i,j})) + (1-y_{i,j})\log(1-\hat{y}_{i,j})) \tag{4.2}$$

where $N$, and $C$ represent the total number of observations, and the total number of classes, respectively. Hence, $\hat{y}_{i,j}$ is a binary indicator (0 or 1) for whether an observation $i$ belongs to class $j$, and $y_{i,j}$ is the predicted probability that observation $i$ belongs to class $j$. Each target $y_{i,j}$ may have multiple positive values since a CT slice may contain multiple ICH subtypes. The weight assigned to each class is denoted by $w_j$, with a weight of 2 assigned to the ANY class, and a weight of 1 assigned to all other classes [14].

However, given the imbalanced nature of the data due to the significant rarity of ICH subtypes across all three datasets, we propose modifying the weighting scheme to place additional emphasis on successfully classifying ICH subtypes. Specifically, a weight of 30 is assigned to positive samples of the EDH class and 6 to all other classes. Since EDH is very rare across all three datasets, it was assigned a greater positive weight, which indicates that misclassifying an EDH sample has a significantly higher penalty than misclassifying other ICH subtypes.

**Training procedure–** This work uses a two-step process for training the proposed model. *Step-I:* the ImageNet pre-trained vision subnetwork is fine-tuned for 20 epochs using 2D CT scan slices with a batch size of 128 to allow the vision network to adapt to the target domain, as shown in Fig. 4.10-(a). The vision network achieved its lowest validation loss score on epoch 9. *Step-II:* A view transformation is applied that reshapes the input dimensions to a sequence of slices instead of a sequence of CT scans. The vision network's trainable parameters are frozen to avoid overfitting when the learner subnetwork is integrated with it. This also helps the model preserves its ability to detect an information-rich set of primitive features from the input CT scan slices. The integrated model–CNN-BiLSM is trained for additional epochs to fully adjust the parameters of the learner subnetwork, while the parameters for the vision network remain unchanged. Fig. 4.10-(b) shows its training progress, where the lowest validation loss score is found to be at the 16th epoch.

**Optimizer–** During the training process, the model's parameters are adjusted in such a way that the loss function is minimized, leading to better predictions. To accomplish this, an AdamW optimizer is used is used to iteratively update these parameters based on the gradients of the loss function with respect to the parameters. The AdamW optimizer extends the Adaptive Moment Estimation (Adam) optimizer by incorporating a weight decay regularization term. AdamW main-

tains the benefits of Adam's adaptive learning rates, while simultaneously ensuring that weight decay has a more pronounced impact on the optimization process. The incorporation of weight decay penalizes large parameter values, which can lead to the model fitting the noise in the training data rather than learning the underlying patterns. The AdamW optimizer is mathematically defined in (4.3).

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t, \tag{4.3}$$

where $w_t$, $\hat{m}_t$, and $\hat{v}_t$ is the weight, and estimates of the first and second moments of the gradients, respectively at time $t$. Hence, $\eta$ is the learning rate, and $\epsilon$ is a small constant to prevent division by zero [73].

**Learning rate scheduler–** A one-cycle learning rate scheduler is used to optimize the training process. The scheduler is designed to dynamically adjust the learning rate throughout the training epochs. By intelligently controlling the learning rate, the scheduler aims to strike a balance between rapid convergence and avoidance of overshooting the optimal solution. The one-cycle scheduler utilizes a cosine annealing strategy with a minimum learning rate of $1 \times 10^{-5}$ and a maximum learning rate of $3 \times 10^{-4}$, as defined in (4.4). At the beginning of the training process, when the model's parameters are far from optimal, a higher learning rate encourages the model to make more significant parameter adjustments and reduces the risk of stagnating at a suboptimal solution in the optimization space. Near the end of the training process, the learning rate is gradually decreased as it approaches the vicinity of the global optimal solution so that the model parameter updates become smaller and more controlled.

$$\eta_t = \eta_{min} + \frac{1}{2}\left(\eta_{max} - \eta_{min}\right)\left(1 + \cos\left(\frac{T_{cur}}{T_{total}}\pi\right)\right), \tag{4.4}$$

where $\eta_t$, $\eta_{min}$, and $\eta_{max}$ stand for the learning rate at iteration $t$, the minimum learning rate, and the maximum learning rate, respectively. $T_{cur}$, $T_{total}$, and $\cos$ is the current training iteration, the total number of training iterations, and the cosine function, respectively [74].

### 4.1.5 Grad-Cam Visualizations

Grad-CAM visualization is a technique used to visualize and comprehend the significant regions of a CT scan slice that contribute to the prediction made by a CNN. The process involves computing importance weights for different regions of the feature maps and generating a heatmap that highlights the most influential areas in the CT scan slice for identifying the target class.

First, the gradient of the target class score, $y^c$, with respect to the feature maps is computed using backpropagation. Let $\frac{\partial y^c}{\partial \mathbf{A}^k}$ represent the gradient of $y^c$ with respect to the $k$-th feature map. The importance weight, $\alpha_k^c$, of the $k$-th feature map in classifying the target class $c$ is calculated as the spatial average of the gradients, as parameterized in (4.5)

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{4.5}$$

where $\mathbf{A}_{ij}^k$ denotes the activation value at position $(i, j)$ in the $k$-th feature map, and $Z$ is the spatial dimensionality of the feature maps.

Next, the importance-weighted feature maps, $\mathbf{L}_c^k$, for the target class $c$ are computed by element-wise multiplication between the importance weight and the corresponding feature map, as computed in (4.6).

$$\mathbf{L}_c^k = \alpha_k^c \cdot \mathbf{A}^k. \tag{4.6}$$

The weighted feature maps are then summed along the channel dimension to obtain a single heatmap, $\mathbf{H}_c$, that highlights the regions of the input CT scan slice most influential for the target class $c$, as defined in (4.7).

$$\mathbf{H}_c = \sum_k \mathbf{L}_c^k. \tag{4.7}$$

To emphasize the most important regions, an element-wise ReLU operation is applied to the heatmap to generate the transformed heatmap, $\mathbf{H}_c^+$, as defined in (4.8).

$$\mathbf{H}_c^+ = \max(\mathbf{H}_c, 0). \tag{4.8}$$

The normalized heatmap, $\mathbf{H}_c^{norm}$, is obtained by scaling the values of the heatmap between 0 and 1, as expressed in (4.9).

$$\mathbf{H}_c^{norm} = \frac{\mathbf{H}_c^+ - \min(\mathbf{H}_c^+)}{\max(\mathbf{H}_c^+) - \min(\mathbf{H}_c^+)}. \tag{4.9}$$

The normalized heatmap is upsampled to match the dimensions of the original input CT scan slice. Finally, the heatmap is overlayed with the input CT scan slice, where the heatmap values determine the intensity of the color to visualize the regions most relevant to the target class [75, 76].

## 4.2 Experimental Analysis

### 4.2.1 Environment

The proposed solution is developed using Python 3.10 and its open-source native libraries along with PyTorch 1.13. The model development, training, and testing are carried out on a system with an AMD Epyc 7413 processor with base clock speed of 2.65 GHz, and an NVIDIA A100 Tensor Core GPU has 6,912 CUDA cores and 40 GB of HBM2 VRAM. The computational resources have been generously provided by the Digital Research Alliance of Canada.

### 4.2.2 Evaluation Metrics

To evaluate the performance of the model, the following metrics are used: accuracy (4.10), sensitivity (4.11), specificity (4.12), precision (4.13), and AUC. By employing these metrics, the per-

formance of the model can be thoroughly evaluated, allowing for comparisons and assessments of its effectiveness in classifying ICH.

Accuracy is the proportion of correctly classified samples, as in (4.10).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4.10}$$

where TP refers to the number of true positives, which are the samples with ICH that are correctly predicted by the model. TN refers to the number of true negatives, which are the samples without ICH that are correctly identified by the model. FP refers to the number of false positives, which are the samples that are incorrectly classified as having ICH. Finally, FN refers to the number of false negatives, which are the samples with ICH that are incorrectly classified as not having hemorrhages by the model.

Sensitivity is the proportion of true positive samples that are correctly identified by the model, as defined in (4.11). A high sensitivity indicates that the model is effective at identifying positive cases, which can be crucial in situations where false negatives are undesirable, such as medical diagnoses.

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \tag{4.11}$$

Specificity is a complementary metric to sensitivity, indicating the proportion of true negative samples correctly identified by the model, as defined in (4.12). High specificity indicates that the model is effective at avoiding false alarms and is particularly important when false positives have significant consequences, such as in medical diagnostics where unnecessary treatments might be administered.

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{4.12}$$

Precision focuses on how well the model performs among the instances it predicts as positive. It indicates the proportion of correctly predicted positive cases out of all cases predicted as positive

by the model. Precision is particularly important when the consequences of false positives are high. For example, in medical testing, a high precision would imply that when the model predicts a positive case, it is highly likely to be accurate. High precision also helps minimize unnecessary interventions or treatments.

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{4.13}$$

On the other hand, the area under the curve, a.k.a. AUC represents the area under the receiver operating characteristic (ROC) curve. An ROC curve is a graphical representation of the performance of a binary classification model that plots the trade-off between its sensitivity and specificity at various decision thresholds. Besides, weighted multi-label binary cross entropy with logits loss defined in (4.2) is also used as one of the evaluation metrics following the RSNA competition standards.

### 4.2.3 Performance Analysis

**Table 4.2:** Performance of the proposed model on RSNA validation set, CQ500 dataset, and PhysioNet datasets using the loss function defined in (4.2). The abbreviations BCE Loss and Inf. Time refer to weighted multi-label binary cross entropy with logits loss and inference time, respectively.

| Dataset | Evaluation Metric | Types of ICH | | | | | | Average | BCE Loss | Inf. Time |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EDH | IPH | IVH | SAH | SDH | ANY | | | |
| **RSNA** | Accuracy | 0.9955 | 0.9834 | 0.9897 | 0.9742 | 0.9689 | 0.9600 | 0.9786 | 0.0501 | 11.7 ms |
| | Sensitivity | 0.2531 | 0.7987 | 0.8537 | 0.6905 | 0.7438 | 0.8722 | 0.8036 | | |
| | Specificity | 0.9982 | 0.9925 | 0.9946 | 0.9885 | 0.9841 | 0.9746 | 0.9891 | | |
| | Precision | 0.3481 | 0.8394 | 0.8473 | 0.7523 | 0.7606 | 0.8509 | 0.8151 | | |
| | AUC | 0.9741 | 0.9909 | 0.9962 | 0.9813 | 0.9810 | 0.9869 | 0.9851 | | |
| **CQ500** | Accuracy | 0.9954 | 0.9819 | 0.9889 | 0.9727 | 0.9656 | 0.9577 | 0.9770 | - | 19.3 ms |
| | Sensitivity | 0.2065 | 0.7674 | 0.8298 | 0.6602 | 0.6682 | 0.8569 | 0.7714 | | |
| | Specificity | 0.9983 | 0.9924 | 0.9946 | 0.9885 | 0.9858 | 0.9745 | 0.9893 | | |
| | Precision | 0.2646 | 0.7989 | 0.8440 | 0.6838 | 0.7024 | 0.8677 | 0.8122 | | |
| | AUC | 0.9708 | 0.9750 | 0.9927 | 0.9721 | 0.9777 | 0.9780 | 0.9777 | | |
| **PhysioNet** | Accuracy | 0.9952 | 0.9805 | 0.9875 | 0.9709 | 0.9651 | 0.9554 | 0.9758 | - | 12.5 ms |
| | Sensitivity | 0.1795 | 0.7338 | 0.7600 | 0.6189 | 0.6515 | 0.8118 | 0.7315 | | |
| | Specificity | 0.9982 | 0.9926 | 0.9955 | 0.9887 | 0.9864 | 0.9792 | 0.9904 | | |
| | Precision | 0.2786 | 0.7699 | 0.7754 | 0.6626 | 0.6739 | 0.8343 | 0.6852 | | |
| | AUC | 0.9396 | 0.9872 | 0.9942 | 0.9764 | 0.9774 | 0.9809 | 0.9759 | | |

**Table 4.3:** Performance of the proposed model on RSNA validation set, CQ500 dataset, and PhysioNet datasets. The loss function defined in (4.2) is modified by assigning a weight of 30 to *positive* samples of the EDH class and 6 to all other classes. The abbreviations BCE Loss and Inf. Time refer to weighted multi-label binary cross entropy with logits loss and inference time, respectively.

| Dataset | Evaluation Metric | Types of ICH | | | | | | Average | BCE Loss | Inf. Time |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EDH | IPH | IVH | SAH | SDH | ANY | | | |
| **RSNA** | Accuracy | 0.9346 | 0.9840 | 0.9897 | 0.9740 | 0.9647 | 0.9562 | 0.9692 | 0.0984 | 11.7 ms |
| | Sensitivity | 0.8468 | 0.9644 | 0.9861 | 0.9479 | 0.9665 | 0.9907 | 0.9744 | | |
| | Specificity | 0.9369 | 0.9871 | 0.9919 | 0.9774 | 0.9667 | 0.9528 | 0.9689 | | |
| | Precision | 0.8598 | 0.9658 | 0.9868 | 0.9497 | 0.9665 | 0.9868 | 0.9763 | | |
| | AUC | 0.9404 | 0.9888 | 0.9946 | 0.9838 | 0.9865 | 0.9905 | 0.9808 | | |
| **CQ500** | Accuracy | 0.8781 | 0.9792 | 0.9840 | 0.9646 | 0.9583 | 0.9505 | 0.9525 | - | 19.3 ms |
| | Sensitivity | 0.8768 | 0.9744 | 0.9938 | 0.8992 | 0.9716 | 0.9940 | 0.9724 | | |
| | Specificity | 0.8781 | 0.9794 | 0.9836 | 0.9679 | 0.9574 | 0.9433 | 0.9513 | | |
| | Precision | 0.8769 | 0.9751 | 0.9929 | 0.9062 | 0.9699 | 0.9898 | 0.9756 | | |
| | AUC | 0.9357 | 0.9892 | 0.9942 | 0.9637 | 0.9836 | 0.9895 | 0.9760 | | |
| **PhysioNet** | Accuracy | 0.9059 | 0.9536 | 0.9915 | 0.9747 | 0.9659 | 0.9585 | 0.9584 | - | 12.5 ms |
| | Sensitivity | 0.8426 | 0.9434 | 0.9849 | 0.9418 | 0.9629 | 0.9910 | 0.9699 | | |
| | Specificity | 0.9061 | 0.9541 | 0.9918 | 0.9764 | 0.9661 | 0.9531 | 0.9577 | | |
| | Precision | 0.8507 | 0.9441 | 0.9855 | 0.9459 | 0.9632 | 0.9856 | 0.9695 | | |
| | AUC | 0.9244 | 0.9756 | 0.9939 | 0.9800 | 0.9833 | 0.9894 | 0.9744 | | |

**RSNA–** Since the ground truth labels for the test set are not provided in the RSNA dataset, the model's performance on the test set can not be assessed using the evaluation metrics stated in Section 4.2.2. Instead, the model is assessed using the validation set, and the results are tabulated in Table 4.2. In this case, the model uses the loss function defined in (4.2). The model obtains average accuracy, sensitivity, specificity, precision, and AUC scores of 0.9786, 0.8036, 0.9891, 0.8151, and 0.9852, respectively. The model has the most difficulty with identifying the EDH class as it only obtains a sensitivity of 0.2531. This is likely because this class is severely underrepresented in the class distribution of the dataset. The proposed model achieves excellent AUC scores of 0.9741, 0.9909, 0.9962, 0.9813, 0.9810, and 0.9869 for the subtypes EDH, IPH, IVH, SAH, SDH, and ANY, respectively. The deep learning model, enriched with comprehensive data augmenta-

tions, yields a significant improvement in performance, as evidenced by a 25.5% reduction in the weighted multi-label logarithmic score from 0.0672 to 0.0501.

Table 4.3 shows the performance of the proposed model when modifying the weighting scheme by assigning a weight of 30 to *positive* samples of the EDH class and 6 to all other classes. This modification enables the model to achieve a notable increase in average sensitivity on the RSNA dataset, rising from 0.8036 to 0.9744, while experiencing only a slight reduction in specificity, from 0.9891 to 0.9744.

In the context of medical emergencies, prioritizing higher sensitivity is generally preferred over specificity. Sensitivity refers to the ability of the model to correctly identify true positive cases, i.e., correctly detecting patients with ICH. Having a higher sensitivity in medical emergencies is crucial because it reduces the likelihood of missing critical cases. In the scenario of ICH, identifying these cases promptly is crucial as timely medical attention and treatment can significantly impact patient outcomes and potentially save lives. While specificity is also essential, it measures the ability of the model to correctly identify true negative cases, i.e., correctly classifying patients without ICH as not having the condition. A slightly lower specificity is generally acceptable in medical emergencies because the primary concern is to identify true positive cases effectively.

While this modification is likely preferable in clinical practice, it is suboptimal for optimizing results in the RSNA competition where the weighted multi-label logarithmic loss evaluation metric is prioritized. The positive weighting scheme, while beneficial for clinical practice, leads to a higher test loss score of 0.0984 compared to 0.0501 without the positive weighting scheme in the competition setting.

The predictions on the test set are uploaded to the RSNA competition site to compute the model's performance in terms of weighted multi-label logarithmic loss and to compare it with the existing solutions. The proposed model achieves a weighted multi-label logarithmic loss of 0.0501 (cf. Table 4.4). It reduces the weighted multi-label logarithmic loss by 6.2% in comparison to the baseline model introduced by Ngo *et al.* while being significantly more computationally efficient. Their solution involves using seven consecutive slices during inference to generate a single prediction for the center slice whereas the proposed model only loads each image once.

**Table 4.4:** Performance Comparison of Various Models on RSNA Test set using the loss function defined in (4.2). "–": Data Not Available, DA: Data Augmentation

| Model Name | Loss | % of Improvement | Inference Time |
|---|---|---|---|
| CNN-GRU [12] | 0.0659 | 23.4 ↓ | – |
| CNN Ensemble [56] | 0.0548 | 2.6 ↓ | – |
| CNN + Axial Fusion [45] | 0.0534 | Baseline | – |
| CNN Ensemble + ViT [49] | 0.0705 | 32.0 ↓ | – |
| CNN-BiLSTM [48] | 0.0522 | 2.2 ↑ | – |
| CNN-LSTM [77] | 0.0753 | 41.0 ↓ | – |
| CNN-BiLSTM w/o DA (ours) | 0.0672 | 25.8 ↓ | 11.7 ms |
| CNN-BiLSTM w/t DA (ours) | 0.0501 | 6.2 ↑ | 11.7 ms |

The baseline model only uses information from six adjacent slices for class prediction of each slice and does not consider long-term sequential dependencies. In contrast, the proposed solution uses sequential information from all slices in a CT scan to make a more accurate prediction.

The proposed model has an average inference speed of only 11.7 ms per CT scan. Unfortunately, the other studies in Table 4.4 do not record any timing analysis. It can be deduced that the proposed solution offers the lowest inference time since the EfficientNetV2 contains fewer computations than the architectures used by other studies. This is because the EfficientNetV2-Small architecture contains depthwise separable convolutions and squeeze-and-excitation blocks to reduce the number of trainable parameters [71]. For example, the EfficientNetV2-Small architecture used in this study contains 21.5 million parameters whereas the ResNet architectures used by Ngo *et al.* contains 25.6 million parameters.

**CQ500–** Using the loss function described in (4.2), the model obtains average accuracy, sensitivity, specificity, precision, and AUC scores of 0.9770, 0.7714, 0.9893, 0.8122, and 0.9777, respectively. The model has the most difficulty with identifying the EDH class as it only obtains a sensitivity of 0.2065. This is likely because this class is severely underrepresented in the class distribution of the RSNA training dataset. Consequently, the proposed model struggles to adapt its parameters to discern the unique characteristics of EDH, resulting in a reduced sensitivity score for EDH on the CQ500 dataset. The proposed model achieves excellent AUC scores of 0.9708,

0.9750, 0.9927, 0.9721, 0.9777, and 0.9780 for the subtypes EDH, IPH, IVH, SAH, SDH, and ANY, respectively.

Table 4.3 presents the results of the proposed model's performance when altering the weighting approach. Specifically, it assigns a weight of 30 to positive samples in the EDH class and a weight of 6 to all other classes. This adjustment allows the model to significantly improve its average sensitivity on the CQ500 dataset, increasing from 0.7714 to 0.9724, while only slightly decreasing specificity from 0.9893 to 0.9513.

**Table 4.5:** Performance Comparison of Various Models on CQ500 dataset using the loss function defined in (4.2) with a weight of 30 assigned to *positive* samples of the EDH class and 6 to all other classes. "–": Data Not Available.

| Model Name | Sensitivity | Specificity | Precision | AUC | Inf. Time |
|---|---|---|---|---|---|
| CNN + RF [19] | 0.942 | 0.710 | – | 0.942 | – |
| Intensity and shape features + KNN[52] | 0.969 | 0.947 | – | – | – |
| RoLo [78] | 0.943 | 0.856 | – | 0.957 | – |
| CNN-ELM [79] | 0.953 | 0.977 | 0.963 | – | – |
| CNN-BiLSTM (ours) | 0.972 | 0.951 | 0.956 | 0.960 | 19.3 ms |

Although the model was not trained directly on the CQ500 dataset, the model was able to effectively learn ICH representation and retain high performance on unseen data. The CQ500 dataset differs from the RSNA dataset in terms of the CT scanner used, CT slice thickness, and patient characteristics. As presented in Table 4.5, the proposed model outperforms state-of-the-art methods in terms of sensitivity and AUC while maintaining a competitive precision and specificity.

The proposed solution overcomes a common limitation of other state-of-the-art studies on the CQ500 dataset in Table 4.5 by exploiting the sequential nature of CT scans. The learner subnetwork in the proposed solution is able to learn the temporal dependencies of slices in each CT scan to enhance the model's predictive capabilities. Hence, the model uses information about how an ICH evolves over time across all slices in a CT scan to improve its predictive capabilities. In contrast, the other studies treat each slice independently which may cause them to miss out on valuable information embedded in the sequence and have reduced performance.

Chilamkurthy *et al.* developed a CNN, based on the ResNet18 architecture, to extract features that serve as input for an RF model for generating classification predictions. However, an RF model may face difficulties with grasping spatial context from the CNN feature embeddings alone [19].

Raghavendra *et al.* extracted shape and intensity based features from their training dataset. Nature-inspired meta-heuristics algorithms, including BA, GWO, and WOA, were used to select the best set of features. The selected features were then used to train a KNN classifier for ICH subtype classification. However, extracting handcrafted features of ICH may not capture generalized ICH representations as effectively as a CNN that can learn complex spatial and hierarchical features from raw pixels. As a result, this model may struggle if tested in a different clinical setting [52].

Guo *et al.* proposed a weakly supervised approach. In this method, a ResNet18 CNN model was also used for feature extraction which was combined with an attention module. However, weakly supervised approaches generally do not achieve the same level of accuracy as fully supervised methods like a CNN-BiLSTM, where ground truth labels are available for training. The ResNet18 architecture is less sophisticated than the EfficientNetV2-Small architecture used in this work, which may reduce its ability to learn generalized ICH feature representations [78].

Santhoshkumar *et al.* used Tsallis entropy with GOA to identify the ROI. Then, a DenseNet CNN model was combined with an ELM for ICH classification. The use of these image-preprocessing strategies may lead to increased computational complexity and difficulty in fine-tuning hyperparameters. ELM assigns random or fixed weights to input features, and as a result, it may not effectively handle irrelevant or noisy features in CT scan slices [79].

**PhysioNet–** Using the loss function described in (4.2), the model obtains average accuracy, sensitivity, specificity, precision, and AUC scores of 0.9758, 0.7315, 0.9904, 0.6852, and 0.9759, respectively. The model has the most difficulty with identifying the EDH class as it only obtains a sensitivity of 0.1795. This is likely because this class is severely underrepresented in the class distribution of the RSNA training dataset. Hence, the proposed model encounters obstacles in adjusting its parameters to effectively capture the distinctive features of EDH, leading to a reduced

sensitivity score for EDH in the PhysioNet dataset. The proposed model achieves excellent AUC scores of 0.9396, 0.9872, 0.9942, 0.9764, 0.9774, and 0.9809 for the subtypes EDH, IPH, IVH, SAH, SDH, and ANY, respectively. The proposed model has an inference time of 12.5 ms for processing an entire CT scan.

Table 4.3 shows the performance of the proposed model when modifying the weighting scheme by assigning a weight of 30 to *positive* samples of the EDH class and 6 to all other classes. This modification enables the model to achieve a notable increase in average sensitivity on the PhysioNet dataset, rising from 0.7315 to 0.9699, while experiencing only a slight reduction in specificity, from 0.9904 to 0.9577.

Since the dataset contains masks that delineate the ICH regions, all reviewed studies have developed segmentation models for detecting ICH. To the best of our knowledge, this is the only study to validate the performance of a classification model on the PhysioNet dataset.

**Grad-CAM–** This study aims to address a significant limitation of DL models, which often face criticism for being ”black box” algorithms. These algorithms tend to generate predictions that lack explainability due to their complex internal workings. To overcome this limitation, Grad-CAM visualizations are used to highlight regions in CT scan slices that have the greatest impact on the final predictions made by the mode. This visualization technique enables radiologists to assess the accuracy of the model's classification predictions. In cases where the model misclassifies, radiologists can examine whether the model is giving undue importance to specific aspects of the CT scan slice. Consequently, appropriate measures can be taken to mitigate this issue. The ability to detect and interpret misclassifications is a crucial step toward developing a system that is both correctable and reliable. Moreover, this technique allows for the visual localization of ICH without requiring pixel-level annotations labelled by radiologists to train the model [50].

Fig. 4.12 presents Grad-Cam visualizations that identify regions with CT scan slices with the highest probability of ICH presence.

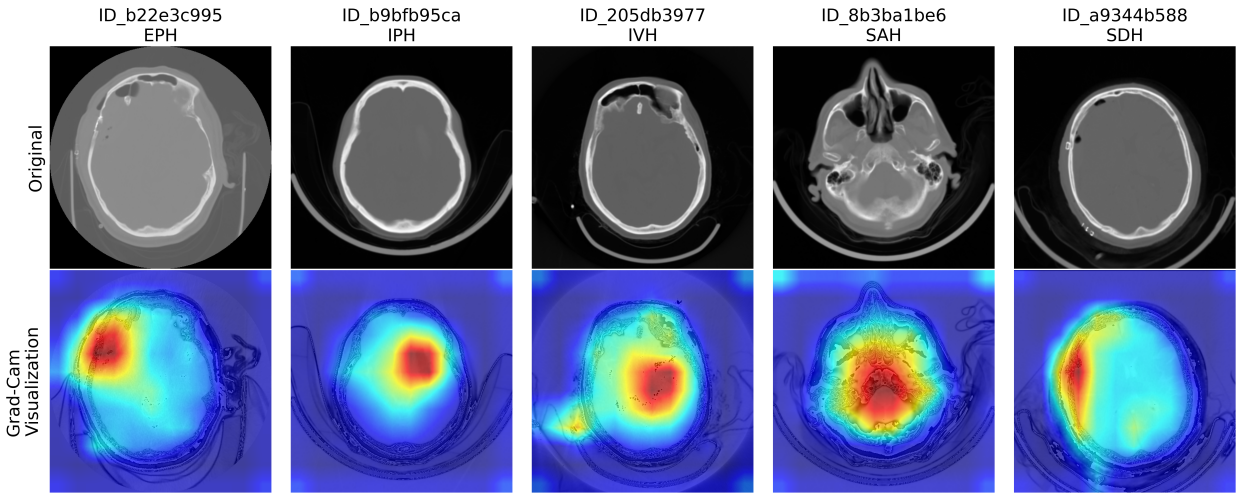



**Figure 4.12:** Grad-Cam visualizations showcase ICH subtypes for every CT scan slice. These visualizations effectively pinpoint regions within the slice where the DL model identifies the highest probability of ICH presence. The highlighted areas in red signify a higher likelihood of ICH, whereas the regions in blue indicate a lower probability.

## 4.3 Conclusion

ICH is a life-threatening condition that requires immediate medical attention. Rapid and accurate diagnosis of ICH is essential for initiating timely treatment, which can significantly impact patient outcomes and improve survival rates. A DL model holds the potential to be used as a triage tool to flag potential ICH cases for further examination by a trained radiologist.

The proposed CNN-BiLSTM classification framework has been developed to automatically identify ICH in CT scan images. Windowing is used as an image preprocessing technique to enhance contrast and highlight potential abnormalities while data augmentations are introduced to increase the diversity of the training dataset. The Vision Network employs a 2-D CNN to learn feature representations of ICH while a BiLSTM network captures sequential patterns among CT scan slices. The trained model is evaluated on RSNA test set, CQ500, and PhysioNet-ICH datasets. Key metrics like sensitivity, specificity, precision, AUC, and weighted logarithmic loss are used to assess its ability to identify ICH instances. Grad-Cam visualization was then used to highlight the regions of the CT scan slices that contribute the most in generating its final classification predictions.

The proposed CNN-BiLSTM classification framework demonstrated excellent performance and generalizability on three datasets. The model achieves an average AUC of 0.9851, 0.9777, and 0.9757 on the RSNA, CQ500, and PhysioNet datasets. Notably, the proposed model achieves a 6.2% reduction in the weighted multi-label logarithmic evaluation metric on the RSNA dataset compared to the baseline model proposed by Ngo *et al.* that combines a CNN with axial fusion [45]. In addition, it outperforms state-of-the-art methods on the CQ500 dataset in terms of sensitivity, and AUC while maintaining competitive precision and specificity performance.

One of the key strengths of this framework is its efficiency in terms of computation time. During inference, the model only required 11.7 ms, 12.5 ms, and 19.3 ms to process an entire CT scan on the RSNA, CQ500, and PhysioNet datasets, respectively. This fast inference time is vital, especially in real-time applications, where timely responses can be potentially life-saving.

The model with the weighted multi-label logarithmic loss defined in (4.2) obtains comparatively lower sensitivity performance than specificity since ICH is underrepresented across all three datasets. The model obtains average sensitivity scores of 0.8038, 0.7714, and 0.7315 compared to average specificity scores of 0.9891, 0.9893, and 0.9904 on the RSNA, CQ500, and PhysioNet datasets respectively. To mitigate this limitation, the loss function defined in (4.2) is modified by assigning a weight of 30 to *positive* samples of the EDH class and 6 to all other classes. This effectively bolstered the sensitivity scores to 0.9744, 0.9724, and 0.9699 while only causing a slight decrease in specificity scores of 0.9689, 0.9513, and 0.9577.

## 4.4 Future Directions

This study employed the EfficientNetV2-Small architecture for learning feature representations of ICH. However, future research could explore the use of larger architectures such as EfficientNetV2-Medium or EfficientNetV2-Large. By increasing the model's capacity, it can better leverage the available training data and potentially enhance its performance. Larger architectures contain more parameters which may the model to capture more intricate patterns and features. With this in mind,

it is important to also consider the computational resources necessary for training and deploying a larger model.

Grad-CAM visualizations were used to identify the regions of CT scan slices that significantly contributed to the model's predictions. However, the analysis did not involve identifying areas where the model may have made errors. To address this, it would have been valuable to involve experienced radiologists in reviewing Grad-CAM visualizations for cases of false positives or false negatives. Radiologists' expertise and feedback can enhance the model's performance and uncover potential areas of improvement. Establishing a feedback loop between the model and radiologists can lead to continuous refinement and bolster the system's overall reliability.

In the study, the loss function was modified to prioritize the accurate classification of under-represented ICH subtypes across the three datasets. An alternative approach that could have been explored is oversampling the ICH samples to address the class imbalance. By increasing the number of training examples for ICH, the model can become more familiar with the specific features and patterns associated with ICH. Oversampling the minority class provides the model with a more balanced distribution of samples during training which can lead to improved generalization and sensitivity towards ICH.

# Chapter 5

# An Improved CNN-BiLSTM Model with Multi-head Attention for Intracranial Hemorrhage Classification

## 5.1 Overview

This chapter proposes incorporating a multi-head attention mechanism into the Learner Subnetwork module of the CNN-BiLSTM CT scan classification framework described in Chapter 4. The multi-head attention mechanism enables the model to better learn global dependencies and capture long-range interactions within the sequential feature embeddings produced by the BiLSTM. It accomplishes this by concentrating on the most pertinent aspects of the feature embeddings. By directing its attention to these crucial areas, the proposed model demonstrates an im-proved ability to identify patterns associated with ICH, resulting in improved predictive performance. An ablation study is conducted to determine the optimal number of attention heads needed in the multi-head attention mechanism design.

## 5.2 Proposed Solution

The incorporation of a multi-head attention mechanism into the learner subnetwork architecture aims to improve the model's capacity to capture and understand the intricate sequential dependencies present among the various slices of CT scans. As illustrated in Fig. 5.1, the multi-head attention mechanism operates on the hidden states derived from the BiLSTM. These hidden states contain valuable information encapsulating temporal relationships and dependencies among the slices. The multi-head attention mechanism acts as a discerning filter for this temporal information. It enables the model to focus its attention on the most salient and contextually relevant characteristics within the sequence of hidden output states produced by the BiLSTM. This selective attention is pivotal because it allows the model to emphasize and weigh certain slices or temporal patterns more heavily, depending on their significance for the given task. Hence, the multi-head attention mechanism produces an enriched representation of the sequential data which is subsequently channeled into the FC layers. The FC layers serve as the final computational stage where the model uses its learned knowledge to generate final classification predictions.

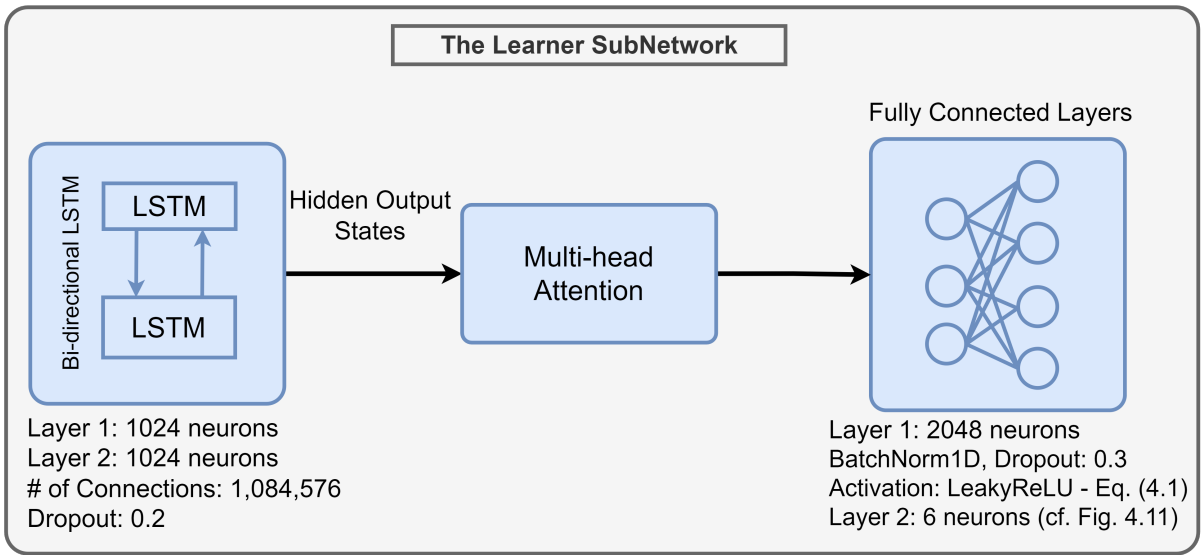



**Figure 5.1:** A flowchart of the modified Learner Subnetwork architecture with the incorporation of a multi-head attention mechanism.

To accommodate varying CT scan lengths resulting from the number of slices taken, masking is employed to efficiently manage input sequences of different sizes without the need for uniformity.

Masking is used to prevent the model from directing its attention to padded positions in the input sequences. This is pivotal as incorporating irrelevant information can introduce noisy or erroneous attention patterns. By systematically masking out padded positions, the model can better focus on the valuable content within each sequence. Furthermore, masking contributes to optimizing computational efficiency during both training and inference. By allowing the model to disregard padded slices, it eliminates the need to process and attend to every position in input sequences, thus reducing unnecessary computational overhead.

To enhance the overall performance of the system, a thorough ablation study is conducted that focuses on fine-tuning the number of attention heads used in the multi-head attention mechanism. Specifically, three different configurations were tested, each with a varying number of attention heads — 4, 8, and 16. After analyzing the performance outcomes, the number of attention heads that yields the highest overall performance was identified.

Furthermore, to address the class imbalance problem, an additional experiment is then conducted using the optimal number of attention heads, and the loss function defined in (4.2) with a weight of 30 assigned to positive samples of the EDH class and 6 to all other classes. This class weighting scheme helps to ensure that the model places a higher emphasis on correctly identifying instances of ICH subtypes due to their rarity across all three benchmark datasets.

It is essential to highlight that these experiments and analyses are carried out using a consistent procedure and set of hyperparameters that have been detailed in Section 4.1 – Proposed Solution. This consistency ensures that the results obtained can be reliably compared and that any observed improvements or changes can be attributed to the specific variations being tested, such as the number of attention heads or the class weighting scheme.

## 5.3 Experimental Analysis

Based on the experimental results in Tables 5.1- 5.3, the number of attention heads has a marginal impact on the performance evaluation metrics. However, using 8 attention heads for the multi-head attention mechanism exhibits slightly better performance. In Table 5.1, using 8 attention

heads leads to a lower weighted log loss score of 0.0482 for the RSNA dataset compared to 0.0488 and 0.0484 when using 4 and 16 attention heads, respectively. In Table 5.2, using 8 attention heads leads to an average AUC of 0.9797 for the CQ500 dataset compared to 0.9795 and 0.9796 when using 4 and 16 attention heads, respectively. In Table 5.3, using 8 attention heads leads to an average AUC of 0.9778 for the PhysioNet dataset compared to 0.9771 and 0.9776 when using 4 and 16 attention heads, respectively.

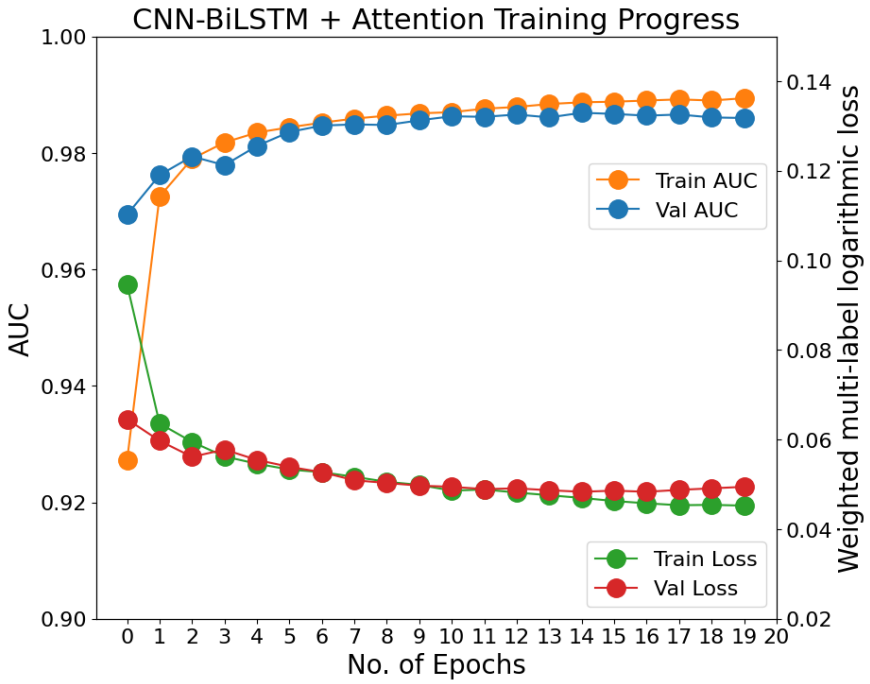



**Figure 5.2:** Training progress of the integrated CNN-BiLSTM model with the inclusion of 8 attention heads. During the training of the CNN-BiLSTM, the vision network's parameters are not updated, i.e., kept frozen.

Since the inclusion of the multi-head attention mechanism in the learner subnetwork does not impact the vision network architecture, it is not necessary to retrain the vision network. The training progress of the integrated CNN-BiLSTM model, featuring the optimal number of 8 attention heads, is visually represented in Fig. 5.2. Upon introducing the multi-head attention mechanism, the validation loss scores exhibit an initial increase during the first four epochs of training, as compared to the training progress of the model without the multi-head attention mechanism, as depicted in Fig. 4.10-(b). This phenomenon can be attributed to the initial learning challenges

faced by the multi-head attention mechanism in effectively attending to various input sequences. However, as the training proceeds, the multi-head attention mechanism gradually becomes more effective in learning relevant temporal information, resulting in lower validation loss scores after the fifth epoch. The CNN-BiLSTM model with 8 attention heads achieves its lowest validation loss on epochs 15 and 17 of 0.0479 and its highest validation AUC score on epoch 15 of 0.9869.

**Table 5.1:** Performance of the proposed model using 4, 8, and 16 attention heads on RSNA validation set while using the loss function defined in (4.2). The abbreviations BCE Loss and Inf. Time refer to weighted multi-label binary cross entropy with logits loss and inference time, respectively.

| **Heads** | **Evaluation Metric** | **Types of ICH** | | | | | | **Ave-rage** | **BCE Loss** | **Inf. Time** |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EDH | IPH | IVH | SAH | SDH | ANY | | | |
| 4 | Accuracy | 0.9854 | 0.9842 | 0.9895 | 0.9746 | 0.9679 | 0.9591 | 0.9785 | 0.0488 | 12.0 ms |
| | Sensitivity | 0.2783 | 0.8317 | 0.8638 | 0.7267 | 0.7628 | 0.8816 | 0.8221 | | |
| | Specificity | 0.9981 | 0.9917 | 0.9939 | 0.9871 | 0.9818 | 0.9720 | 0.9878 | | |
| | Precision | 0.3744 | 0.8411 | 0.8780 | 0.7425 | 0.7951 | 0.8891 | 0.8014 | | |
| | AUC | 0.9802 | 0.9954 | 0.9892 | 0.9878 | 0.9731 | 0.9820 | 0.9863 | | |
| 8 | Accuracy | 0.9853 | 0.9842 | 0.9895 | 0.9747 | 0.9680 | 0.9592 | 0.9785 | 0.0482 | 12.0 ms |
| | Sensitivity | 0.2819 | 0.8309 | 0.8636 | 0.7270 | 0.7637 | 0.8820 | 0.8223 | | |
| | Specificity | 0.9981 | 0.9917 | 0.9940 | 0.9872 | 0.9818 | 0.9720 | 0.9878 | | |
| | Precision | 0.3487 | 0.8514 | 0.8784 | 0.7465 | 0.7891 | 0.8905 | 0.8018 | | |
| | AUC | 0.9814 | 0.9955 | 0.9907 | 0.9873 | 0.9736 | 0.9828 | 0.9869 | | |
| 16 | Accuracy | 0.9854 | 0.9842 | 0.9895 | 0.9746 | 0.9680 | 0.9591 | 0.9785 | 0.0484 | 12.0 ms |
| | Sensitivity | 0.2837 | 0.8320 | 0.8621 | 0.7252 | 0.7638 | 0.8817 | 0.8220 | | |
| | Specificity | 0.9981 | 0.9917 | 0.9940 | 0.9872 | 0.9819 | 0.9720 | 0.9878 | | |
| | Precision | 0.3621 | 0.8558 | 0.8728 | 0.7499 | 0.7935 | 0.8936 | 0.8018 | | |
| | AUC | 0.9821 | 0.9955 | 0.9900 | 0.9878 | 0.9729 | 0.9827 | 0.9868 | | |

**Table 5.2:** Performance of the proposed model using 4, 8, and 16 attention heads on CQ500 dataset while using the loss function defined in (4.2).

| Heads | Evaluation Metric | Types of ICH | | | | | | Average | Inf. Time |
|---|---|---|---|---|---|---|---|---|---|
| | | EDH | IPH | IVH | SAH | SDH | ANY | | |
| 4 | Accuracy | 0.9954 | 0.9823 | 0.9891 | 0.9730 | 0.9665 | 0.9583 | 0.9774 | 19.8 ms |
| | Sensitivity | 0.2208 | 0.7748 | 0.8345 | 0.6663 | 0.6809 | 0.8605 | 0.7778 | |
| | Specificity | 0.9983 | 0.9925 | 0.9946 | 0.9885 | 0.9858 | 0.9745 | 0.9894 | |
| | Precision | 0.3214 | 0.8003 | 0.8570 | 0.6919 | 0.7107 | 0.8708 | 0.8140 | |
| | AUC | 0.9714 | 0.9785 | 0.9942 | 0.9749 | 0.9784 | 0.9798 | 0.9795 | |
| 8 | Accuracy | 0.9954 | 0.9822 | 0.9891 | 0.9731 | 0.9666 | 0.9584 | 0.9774 | 19.8 ms |
| | Sensitivity | 0.2118 | 0.7742 | 0.8337 | 0.6673 | 0.6813 | 0.8614 | 0.7782 | |
| | Specificity | 0.9983 | 0.9925 | 0.9946 | 0.9885 | 0.9859 | 0.9745 | 0.9894 | |
| | Precision | 0.2610 | 0.8053 | 0.8543 | 0.6971 | 0.7215 | 0.8714 | 0.8141 | |
| | AUC | 0.9714 | 0.9787 | 0.9944 | 0.9751 | 0.9785 | 0.9800 | 0.9797 | |
| 16 | Accuracy | 0.9954 | 0.9823 | 0.9891 | 0.9730 | 0.9665 | 0.9584 | 0.9775 | 19.9 ms |
| | Sensitivity | 0.2101 | 0.7754 | 0.8360 | 0.6667 | 0.6822 | 0.8614 | 0.7786 | |
| | Specificity | 0.9983 | 0.9925 | 0.9946 | 0.9885 | 0.9858 | 0.9745 | 0.9894 | |
| | Precision | 0.2500 | 0.7993 | 0.8485 | 0.7129 | 0.7109 | 0.8681 | 0.8139 | |
| | AUC | 0.9713 | 0.9787 | 0.9944 | 0.9751 | 0.9785 | 0.9800 | 0.9796 | |

**Table 5.3:** Performance of the proposed model using 4, 8, and 16 attention heads on the PhysioNet dataset using the loss function defined in (4.2).

| Heads | Evaluation Metric | Types of ICH | | | | | | Average | Inf. Time |
|---|---|---|---|---|---|---|---|---|---|
| | | EDH | IPH | IVH | SAH | SDH | ANY | | |
| 4 | Accuracy | 0.9953 | 0.9818 | 0.9890 | 0.9723 | 0.9674 | 0.9585 | 0.9774 | 12.8 ms |
| | Sensitivity | 0.2029 | 0.7605 | 0.8022 | 0.6487 | 0.6871 | 0.8336 | 0.7597 | |
| | Specificity | 0.9982 | 0.9926 | 0.9956 | 0.9887 | 0.9864 | 0.9792 | 0.9904 | |
| | Precision | 0.2784 | 0.7860 | 0.8234 | 0.6851 | 0.7192 | 0.8412 | 0.7084 | |
| | AUC | 0.9379 | 0.9884 | 0.9950 | 0.9782 | 0.9795 | 0.9842 | 0.9771 | |
| 8 | Accuracy | 0.9953 | 0.9820 | 0.9890 | 0.9727 | 0.9680 | 0.9593 | 0.9777 | 12.8 ms |
| | Sensitivity | 0.2065 | 0.7652 | 0.8049 | 0.6551 | 0.6979 | 0.8398 | 0.7662 | |
| | Specificity | 0.9982 | 0.9926 | 0.9956 | 0.9887 | 0.9863 | 0.9792 | 0.9904 | |
| | Precision | 0.2936 | 0.7809 | 0.8242 | 0.6820 | 0.7190 | 0.8503 | 0.7112 | |
| | AUC | 0.9394 | 0.9887 | 0.9953 | 0.9787 | 0.9800 | 0.9850 | 0.9778 | |
| 16 | Accuracy | 0.9953 | 0.9820 | 0.9890 | 0.9726 | 0.9681 | 0.9592 | 0.9777 | 12.8 ms |
| | Sensitivity | 0.2101 | 0.7656 | 0.8018 | 0.6532 | 0.6976 | 0.8381 | 0.7650 | |
| | Specificity | 0.9982 | 0.9926 | 0.9957 | 0.9887 | 0.9864 | 0.9793 | 0.9904 | |
| | Precision | 0.2828 | 0.7813 | 0.8218 | 0.6954 | 0.7242 | 0.8496 | 0.7119 | |
| | AUC | 0.9384 | 0.9886 | 0.9952 | 0.9786 | 0.9799 | 0.9849 | 0.9776 | |

**Table 5.4:** Performance of the proposed model RSNA validation set, CQ500 dataset, and PhysioNet while using 8 attention heads. The loss function defined in (4.2) is modified by assigning a weight of 30 to *positive* samples of the EDH class and 6 to all other classes.

| Dataset | Evaluation Metric | Types of ICH | | | | | | Average | BCE Loss | Inf. Time |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EDH | IPH | IVH | SAH | SDH | ANY | | | |
| **RSNA** | Accuracy | 0.9402 | 0.9859 | 0.9910 | 0.9749 | 0.9664 | 0.9585 | 0.9715 | 0.0947 | 12.0 ms |
| | Sensitivity | 0.8684 | 0.9693 | 0.9896 | 0.9530 | 0.9680 | 0.9914 | 0.9770 | | |
| | Specificity | 0.9425 | 0.9889 | 0.9931 | 0.9781 | 0.9684 | 0.9554 | 0.9712 | | |
| | Precision | 0.8776 | 0.9722 | 0.9899 | 0.9563 | 0.9680 | 0.9879 | 0.9826 | | |
| | AUC | 0.9477 | 0.9917 | 0.9960 | 0.9842 | 0.9862 | 0.9908 | 0.9828 | | |
| **CQ500** | Accuracy | 0.8940 | 0.9829 | 0.9883 | 0.9691 | 0.9631 | 0.9552 | 0.9588 | - | 19.8 ms |
| | Sensitivity | 0.8821 | 0.9779 | 0.9909 | 0.9021 | 0.9715 | 0.9960 | 0.9738 | | |
| | Specificity | 0.8940 | 0.9832 | 0.9882 | 0.9725 | 0.9625 | 0.9484 | 0.9578 | | |
| | Precision | 0.8831 | 0.9784 | 0.9907 | 0.9063 | 0.9707 | 0.9890 | 0.9769 | | |
| | AUC | 0.9387 | 0.9910 | 0.9943 | 0.9677 | 0.9842 | 0.9904 | 0.9777 | | |
| **PhysioNet** | Accuracy | 0.9110 | 0.9557 | 0.9924 | 0.9766 | 0.9679 | 0.9594 | 0.9605 | - | 12.8 ms |
| | Sensitivity | 0.8409 | 0.9503 | 0.9862 | 0.9538 | 0.9670 | 0.9922 | 0.9740 | | |
| | Specificity | 0.9113 | 0.9560 | 0.9926 | 0.9778 | 0.9680 | 0.9540 | 0.9597 | | |
| | Precision | 0.8452 | 0.9508 | 0.9867 | 0.9560 | 0.9672 | 0.9882 | 0.9726 | | |
| | AUC | 0.9334 | 0.9783 | 0.9934 | 0.9838 | 0.9848 | 0.9899 | 0.9773 | | |

**RSNA–** Since the ground truth labels for the test set are not provided in the RSNA dataset, the model's performance on the test set can not be assessed using the evaluation metrics stated in Section 4.2.2. Instead, the model is assessed using the validation set. The results of using varying numbers of attention heads for training the proposed model on the RSNA dataset are tabulated in Table 5.1. In these experiments, the model uses the loss function described in (4.2). Using 8 attention heads, the model obtains average accuracy, sensitivity, specificity, precision, and AUC scores of 0.9785, 0.8223, 0.9878, 0.8018, and 0.9869, respectively. The model has the most difficulty with identifying the EDH class as it only obtains a sensitivity of 0.2819. This is likely because this class is severely underrepresented in the class distribution of the dataset. The proposed model achieves excellent AUC scores of 0.9814, 0.9955, 0.9907, 0.9873, 0.9736, and 0.9828 for the subtypes EDH, IPH, IVH, SAH, SDH, and ANY, respectively. The model requires 12.0 ms to process an entire sequence of CT scans.

Table 5.4 shows the performance of the proposed model while using 8 attention heads when modifying the weighting scheme by assigning a weight of 30 to positive samples of the EDH class and 6 to all other classes. This modification enables the model to achieve a notable increase in average sensitivity on the RSNA dataset, rising from 0.8223 to 0.9770, while experiencing only a slight reduction in specificity, from 0.9894 to 0.9712. However, this drastically increased the weighted multi-label logarithmic score from 0.0482 to 0.0947.

**CQ500–** The results of using varying numbers of attention heads for training the proposed model on the CQ500 dataset are tabulated in Table 5.2. In these experiments, the model uses the loss function described in (4.2). Using 8 attention heads, the model obtains average accuracy, sensitivity, specificity, precision, and AUC scores of 0.9714, 0.9787, 0.9944, 0.9751, and 0.99785, respectively. The model has the most difficulty with identifying the EDH class as it only obtains a sensitivity of 0.2118. This is likely because this class is severely underrepresented in the class distribution of the dataset. The proposed model achieves excellent AUC scores of 0.9714, 0.9787, 0.9944, 0.9751, 0.9785, and 0.9800 for the subtypes EDH, IPH, IVH, SAH, SDH, and ANY, respectively. The model requires 19.8 ms to process an entire sequence of CT scans.

Table 5.4 shows the performance of the proposed model while using 8 attention heads when modifying the weighting scheme by assigning a weight of 30 to *positive* samples of the EDH class and 6 to all other classes. This modification enables the model to achieve a notable increase in average sensitivity on the CQ500 dataset, rising from 0.7782 to 0.9738, while experiencing only a slight reduction in specificity, from 0.9878 to 0.9578.

**PhysioNet–** The results of using varying numbers of attention heads for training the proposed model on the RSNA dataset are tabulated in Table 5.3. In these experiments, the model uses the loss function described in (4.2). Using eight attention heads, the model obtains average accuracy, sensitivity, specificity, precision, and AUC scores of 0.9777, 0.7662, 0.9904, 0.7112, and 0.9778, respectively. The model has the most difficulty with identifying the EDH class as it only obtains a sensitivity of 0.2065. This is likely because this class is severely underrepresented in the class distribution of the RSNA training dataset. The proposed model achieves excellent AUC scores of 0.9384, 0.9887, 0.9953, 0.9787, 0.9800, and 0.9850 for the subtypes EDH, IPH, IVH, SAH, SDH,

and ANY, respectively. The proposed model has an inference time of 12.8 ms for processing an entire CT scan.

Table 5.4 shows the performance of the proposed model while using 8 attention heads when modifying the weighting scheme by assigning a weight of 30 to *positive* samples of the EDH class and 6 to all other classes. This modification enables the model to achieve a notable increase in average sensitivity on the PhsyioNet dataset, rising from 0.7662 to 0.9740, while experiencing only a slight reduction in specificity, from 0.9904 to 0.9597.

## 5.4 Conclusion

In this chapter, a multi-head attention mechanism was integrated into the design of the Learner Subnetwork module of the CNN-BiLSTM CT scan classification framework described in Chapter 4. The primary goal of this attention mechanism is to facilitate the model in better understanding global dependencies and capturing long-range interactions within the sequential feature embeddings generated by the BiLSTM. The multi-head attention mechanism achieves this by selectively focusing on the most relevant aspects of the feature embeddings. This targeted attention allows the proposed model to excel in identifying essential patterns associated with ICH, resulting in a marginal boost in predictive performance.

To fine-tune the multi-head attention mechanism's design, an ablation study is carried out to determine the optimal number of attention heads required. The experimental results indicate that using 8 attention heads for the multi-head attention mechanism is marginally better than using 4 or 16 attention heads. While the multi-head attention mechanism does slightly increase inference time, its improved predictive accuracy can be valuable in clinical settings with sufficient hardware resources to manage the added computational load. The experimental results across three datasets demonstrate the effectiveness of incorporating a multi-head attention mechanism into the CNN-BiLSTM design.

While a multi-head attention mechanism was incorporated after the Bi-LSTM, it may also be beneficial to consider incorporating an attention mechanism into the Vision Network. By incorporating attention into the Vision Network, the model can focus on specific regions or features of the input image that are most relevant to the task at hand. For instance, the inclusion of spatial attention following the final convolutional layer empowers the network to selectively prioritize vital visual cues. Spatial attention has the potential to enhance the model's capacity to extract pertinent features and consequently improve its predictive capabilities.

# Chapter 6

# Conclusion

Accurate and early detection of ICH through the use of DL models has the potential to significantly impact patient outcomes. Timely detection allows for prompt intervention and treatment, leading to improved prognosis and reduced morbidity and mortality rates. The integration of this model into clinical practice can optimize the workflow of radiologists and healthcare providers which may save valuable time and resources.

The proposed classification framework, CNN-BiLSTM, was designed to automatically detect Intracranial Hemorrhage (ICH) in CT scan images. To improve image contrast and identify potential abnormalities, *windowing* was employed as a preprocessing technique, and data augmentations were introduced to diversify the training dataset. The Vision Subnetwork uses a 2-D CNN to learn feature representations of ICH, while the BiLSTM network captures sequential patterns in CT scan slices. The performance of the trained model was evaluated on multiple datasets, including RSNA-ICH test set, CQ500, and PhysioNet-ICH. Various key metrics such as sensitivity, specificity, precision, AUC, and weighted logarithmic loss were used to measure its ability to identify instances of ICH. To gain insights into the model's decision-making process, Grad-Cam visualization was utilized. This technique highlights the regions of the CT scan slices that contribute the most to the final classification predictions.

The experimental results in Chapter 4 demonstrate the exceptional performance and adaptability of the proposed CNN-BiLSTM classification framework across three datasets. Compared to the

baseline model by Ngo *et al.* that combines a CNN with axial fusion, the proposed model achieves a 6.2% reduction in the weighted multi-label logarithmic evaluation metric on the RSNA dataset. Furthermore, on the CQ500 dataset, our model outperforms state-of-the-art methods in sensitivity, precision, and AUC, while maintaining competitive specificity performance. During inference, the model only requires 11.7 ms, 12.5 ms, and 19.3 ms to process an entire CT scan on the RSNA, CQ500, and PhysioNet datasets, respectively. Such rapid inference times are crucial, especially in real-time applications, where quick responses can have life-saving implications. These performance metrics highlight the potential of the model to serve as a reliable tool for radiologists and healthcare professionals in diagnosing intracranial hemorrhage with precision and efficiency.

In Chapter 5, a multi-head attention mechanism is integrated into the Learner Subnetwork architecture to learn global dependencies and capture long-range interactions within the sequential feature embeddings. The attention mechanism allows the model to concentrate on pertinent characteristics and accentuate the most informative aspects of within the CT scan slices. After conducting an ablation study, it was determined that using eight attention heads leads to the best performance.

Overall, the development of DL models for detecting ICH in CT scans holds immense promise for enhancing medical diagnostics and patient care. By harnessing the potential of DL models, the speed and accuracy of ICH detection, lead to enhanced patient outcomes, optimized resource allocation, and ultimately improved healthcare. The research findings and future directions outlined in this study lay the foundation for further advancements in developing DL models for detecting ICH.

## 6.1 Future Directions

The proposed model for detecting and classifying ICH has undergone extensive evaluation using three publicly available datasets. These evaluations have provided valuable insights into the model's performance and its potential application in ICH detection. However, in order to determine its effectiveness in real-world scenarios, it is crucial to deploy the model into clinical settings, such as radiology departments or emergency rooms, to conduct validation studies. It should be em-

phasized that the transition from testing on publicly available datasets to real-world clinical practice can pose several challenges that reduce the model's performance. For example, the publicly available datasets may not fully represent the diversity and complexity of the patient population encountered in clinical practice. Factors such as data acquisition, image quality, and compatibility with existing systems, may limit the practical utility of the proposed model.

Furthermore, the proposed model can serve as a valuable triage tool in assisting medical professionals with prioritizing care for patients. By quickly flagging potential cases of ICH, the model can help assess the urgency and severity of different patient cases so that appropriate resources and treatment can be provided accordingly. In addition, the validation of the model within a clinical workflow allows for the collection of real-time feedback from healthcare professionals. This feedback may offer important insights into the limitations of the proposed model. Subsequently, the proposed model can be iteratively improved to reduce the impact of its limitations and enhance predictive accuracy. Overall, integrating the proposed model into clinical workflow can lead to significant benefits, such as enhanced accuracy and efficiency in ICH detection, improved patient outcomes, and reduced healthcare costs.

Future studies may explore the application of Graph Neural Networks (GNNs) for the automatic detection of ICH. Due to their ability to learn complex relationships in imaging data, GNNs have recently become very popular in the field of medical imaging. ICH detection often requires considering the spatial relationships between different regions within the brain. GNNs can effectively capture these relationships by modeling the brain as a graph, where nodes represent different brain regions and edges represent their connections. By incorporating this structural information into the model, GNNs may be able to detect ICH with a high degree of accuracy by extracting information from the interdependencies between brain regions. Furthermore, GNNs inherently confer interpretability which is crucial for medical imaging tasks. GNNs provide insights into the importance and contribution of different brain regions in the detection of ICH. This quality may enhance the trust and acceptance of the system among healthcare professionals [80].

# Bibliography


[1] N. Hamada and L. B. Zablotska, "New evidence for brain cancer risk after a single paediatric ct scan," *The Lancet Oncology*, vol. 24, no. 1, pp. 2–3, 2023.

[2] A. E. Flanders, L. M. Prevedello, G. Shih, S. S. Halabi, J. Kalpathy-Cramer, R. Ball, J. T. Mongan, A. Stein, F. C. Kitamura, M. P. Lungren *et al.*, "Construction of a machine learning dataset through collaboration: the rsna 2019 brain ct hemorrhage challenge," *Radiology: Artificial Intelligence*, vol. 2, no. 3, p. e190211, 2020.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] J. A. Caceres and J. N. Goldstein, "Intracranial hemorrhage," *Emergency medicine clinics of North America*, vol. 30, no. 3, pp. 771–794, 2012.

[6] G. Xi, R. F. Keep, and J. T. Hoff, "Mechanisms of brain injury after intracerebral haemorrhage," *The Lancet Neurology*, vol. 5, no. 1, pp. 53–63, 2006.

[7] E. M. Hylek and D. E. Singer, "Risk factors for intracranial hemorrhage in outpatients taking warfarin," *Annals of internal medicine*, vol. 120, no. 11, pp. 897–902, 1994.

[8] A. M. Naidech, "Intracranial hemorrhage," *American journal of respiratory and critical care medicine*, vol. 184, no. 9, pp. 998–1006, 2011.

[9] W. D. Freeman and M. I. Aguilar, "Intracranial hemorrhage: diagnosis and management," *Neurologic clinics*, vol. 30, no. 1, pp. 211–240, 2012.

[10] M. Schrag and H. Kirshner, "Management of intracerebral hemorrhage: Jacc focus seminar," *Journal of the American College of Cardiology*, vol. 75, no. 15, pp. 1819–1831, 2020.

[11] I. C. Hostettler, D. J. Seiffge, and D. J. Werring, "Intracerebral hemorrhage: an update on diagnosis and treatment," *Expert review of neurotherapeutics*, vol. 19, no. 7, pp. 679–694, 2019.

[12] P. Kadam, J. Raphael, P. Karale, I. D'silva, and K. Sonawane, "A cnn-rnn based approach for simultaneous detection, identification and classification of intracranial hemorrhage," in *2021 Intl. conf. on Communication information and Computing Technology (ICCICT)*, 2021, pp. 1–6.

[13] J. Broderick, S. Connolly, E. Feldmann, D. Hanley, C. Kase, D. Krieger, M. Mayberg, L. Morgenstern, C. S. Ogilvy, P. Vespa *et al.*, "Guidelines for the management of spontaneous intracerebral hemorrhage in adults: 2007 update: A guideline from the american heart association/american stroke association stroke council, high blood pressure research council, and the quality of care and outcomes in research interdisciplinary working group: The american academy of neurology affirms the value of this guideline as an educational tool for neurologists." *Stroke*, vol. 38, no. 6, pp. 2001–2023, 2007.

[14] X. Wang, T. Shen, S. Yang, J. Lan, Y. Xu, M. Wang, J. Zhang, and X. Han, "A deep learning algorithm for automatic detection and classification of acute intracranial hemorrhages in head ct scans," *NeuroImage: Clinical*, vol. 32, p. 102785, 2021.

[15] M. R. Arbabshirani, B. K. Fornwalt, G. J. Mongelluzzo, J. D. Suever, B. D. Geise, A. A. Patel, and G. J. Moore, "Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration," *NPJ digital medicine*, vol. 1, no. 1, p. 9, 2018.

[16] M. L. Wilson, R. Atun, K. DeStigter, J. Flanigan, K. A. Fleming, S. Horton, S. Kleinert, and S. Sayed, "The lancet commission on diagnostics: advancing equitable access to diagnostics," *The Lancet*, vol. 393, no. 10185, pp. 2018–2020, 2019.

[17] T. Razi, M. Niknami, and F. A. Ghazani, "Relationship between hounsfield unit in ct scan and gray scale in cbct," *Journal of dental research, dental clinics, dental prospects*, vol. 8, no. 2, p. 107, 2014.

[18] P. Parizel, S. Makkat, E. Van Miert, J. Van Goethem, L. Van den Hauwe, and A. De Schepper, "Intracranial hemorrhage: principles of ct and mri interpretation," *European radiology*, vol. 11, pp. 1770–1783, 2001.

[19] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N. G. Campeau, V. K. Venugopal, V. Mahajan, P. Rao, and P. Warier, "Deep learning algorithms for detection of critical findings in head ct scans: a retrospective study," *The Lancet*, vol. 392, no. 10162, pp. 2388–2396, 2018.

[20] T. D. DenOtter and J. Schubert, "Hounsfield unit," 2019.

[21] S. Kamalian, M. H. Lev, and R. Gupta, "Computed tomography imaging and angiography–principles," *Handbook of clinical neurology*, vol. 135, pp. 3–20, 2016.

[22] D. Wu, G. Wang, B. Bian, Z. Liu, and D. Li, "Benefits of low-dose ct scan of head for patients with intracranial hemorrhage," *Dose-Response*, vol. 18, no. 1, p. 1559325820909778, 2020.

[23] B. A. Gross, B. T. Jankowitz, and R. M. Friedlander, "Cerebral intraparenchymal hemorrhage: a review," *Jama*, vol. 321, no. 13, pp. 1295–1303, 2019.

[24] S. Tenny and W. Thorell, "Intracranial hemorrhage," in *StatPearls [Internet]*. StatPearls Publishing, 2022.

[25] J. J. Heit, M. Iv, and M. Wintermark, "Imaging of intracranial hemorrhage," *Journal of stroke*, vol. 19, no. 1, p. 11, 2017.

[26] S. E. Mirvis and K. Shanmuganathan, "Trauma radiology: part iv. imaging of acute craniocerebral trauma," *Journal of Intensive Care Medicine*, vol. 9, no. 6, pp. 305–315, 1994.

[27] H. S. Ivamoto, H. P. Lemos Jr, and A. N. Atallah, "Surgical treatments for chronic subdural hematomas: a comprehensive systematic review," *World neurosurgery*, vol. 86, pp. 399–418, 2016.

[28] D. Rajashekar and J. W. Liang, "Intracerebral hemorrhage," 2020.

[29] M. E. Mayerhoefer, A. Materka, G. Langs, I. Häggström, P. Szczypiński, P. Gibbs, and G. Cook, "Introduction to radiomics," *Journal of Nuclear Medicine*, vol. 61, no. 4, pp. 488–495, 2020.

[30] V. Kumar, Y. Gu, S. Basu, A. Berglund, S. A. Eschrich, M. B. Schabath, K. Forster, H. J. Aerts, A. Dekker, D. Fenstermacher *et al.*, "Radiomics: the process and the challenges," *Magnetic resonance imaging*, vol. 30, no. 9, pp. 1234–1248, 2012.

[31] A. Gudigar, U. Raghavendra, A. Hegde, G. R. Menon, F. Molinari, E. J. Ciaccio, U. R. Acharya *et al.*, "Automated detection and screening of traumatic brain injury (tbi) using computed tomography images: a comprehensive review and future perspectives," *International journal of environmental research and public health*, vol. 18, no. 12, p. 6499, 2021.

[32] D. M. Alawad, A. Mishra, and M. T. Hoque, "Aibh: accurate identification of brain hemorrhage using genetic algorithm based feature selection and stacking," *Machine Learning and Knowledge Extraction*, vol. 2, no. 2, pp. 56–77, 2020.

[33] M. Al-Ayyoub, D. Alawad, K. Al-Darabsah, and I. Aljarrah, "Automatic detection and classification of brain hemorrhages," *WSEAS transactions on computers*, vol. 12, no. 10, pp. 395–405, 2013.

[34] M. Grewal, M. M. Srivastava, P. Kumar, and S. Varadarajan, "Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans," in *IEEE 15th International Sympo. Biomedical Imaging*. IEEE, 2018, pp. 281–284.

[35] J. Ker, S. P. Singh, Y. Bai, J. Rao, T. Lim, and L. Wang, "Image thresholding improves 3-dimensional convolutional neural network diagnosis of different acute brain hemorrhages on computed tomography scans," *Sensors*, vol. 19, no. 9, p. 2167, 2019.

[36] P. Kumaravel, S. Mohan, J. Arivudaiyanambi, N. Shajil, and H. N. Venkatakrishnan, "A simplified framework for the detection of intracranial hemorrhage in ct brain images using deep learning," *Current medical imaging*, vol. 17, no. 10, pp. 1226–1236, 2021.

[37] T. Chan, "Computer aided detection of small acute intracranial hemorrhage on computer tomography of brain," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4-5, pp. 285–298, 2007.

[38] P. D. Chang, E. Kuoy, J. Grinband, B. D. Weinberg, M. Thompson, R. Homo, J. Chen, H. Abcede, M. Shafie, L. Sugrue *et al.*, "Hybrid 3d/2d convolutional neural network for hemorrhage evaluation on head ct," *American Journal of Neuroradiology*, vol. 39, no. 9, pp. 1609–1616, 2018.

[39] W. Kuo, C. Hne, P. Mukherjee, J. Malik, and E. L. Yuh, "Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22 737–22 745, 2019.

[40] J. J. Titano, M. Badgeley, J. Schefflein, M. Pain, A. Su, M. Cai, N. Swinburne, J. Zech, J. Kim, J. Bederson *et al.*, "Automated deep-neural-network surveillance of cranial images for acute neurologic events," *Nature medicine*, vol. 24, no. 9, pp. 1337–1341, 2018.

[41] H. S. Bhadauria, A. Singh, and M. Dewal, "An integrated method for hemorrhage segmentation from brain ct imaging," *Computers & Electrical Engineering*, vol. 39, no. 5, pp. 1527–1536, 2013.

[42] A. Gautam, B. Raman, and S. Raghuvanshi, "A hybrid approach for the delineation of brain lesion from ct images," *Biocybernetics and Biomedical Engineering*, vol. 38, no. 3, pp. 504–518, 2018.

[43] Y.-H. Li, L. Zhang, Q.-M. Hu, H.-W. Li, F.-C. Jia, and J.-H. Wu, "Automatic subarachnoid space segmentation and hemorrhage detection in clinical head ct scans," *International journal of computer assisted radiology and surgery*, vol. 7, pp. 507–516, 2012.

[44] M. D. Hssayeni, M. S. Croock, A. D. Salman, H. F. Al-khafaji, Z. A. Yahya, and B. Ghoraani, "Intracranial hemorrhage segmentation using a deep convolutional model," *Data*, vol. 5, no. 1, p. 14, 2020.

[45] D. T. Ngo, H. H. Pham, T. T. Nguyen, H. T. Nguyen, D. B. Nguyen, and H. Q. Nguyen, "Slice-level detection of intracranial hemorrhage on ct using deep descriptors of adjacent slices," *arXiv preprint arXiv:2208.03403*, 2022.

[46] H. Lee, S. Yune, M. Mansouri, M. Kim, S. H. Tajmir, C. E. Guerrier, S. A. Ebert, S. R. Pomerantz, J. M. Romero, S. Kamalian *et al.*, "An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets," *Nature biomedical engineering*, vol. 3, no. 3, pp. 173–182, 2019.

[47] D. Alis, C. Alis, M. Yergin, C. Topel, O. Asmakutlu, O. Bagcilar, Y. D. Senli, A. Ustundag, V. Salt, S. N. Dogan *et al.*, "A joint convolutional-recurrent neural network with an attention mechanism for detecting intracranial hemorrhage on noncontrast head ct," *Scientific Reports*, vol. 12, no. 1, p. 2084, 2022.

[48] N. T. Nguyen, D. Q. Tran, N. T. Nguyen, and H. Q. Nguyen, "A cnn-lstm architecture for detection of intracranial hemorrhage on ct scans," *medRxiv*, pp. 2020–04, 2020.

[49] Y. Barhoumi, N. C. Bouaynaya, and G. Rasool, "Efficient scopeformer: Towards scalable and rich feature extraction for intracranial hemorrhage detection," *arXiv preprint arXiv:2302.00220*, 2023.

[50] M. Yeo, B. Tahayori, H. K. Kok, J. Maingard, N. Kutaiba, J. Russell, V. Thijs, A. Jhamb, R. V. Chandra, M. Brooks *et al.*, "Evaluation of techniques to improve a deep learning algorithm for the automatic detection of intracranial haemorrhage on ct head imaging," *European Radiology Experimental*, vol. 7, no. 1, p. 17, 2023.

[51] Ö. F. Ertuğrul and M. F. Akıl, "Detecting hemorrhage types and bounding box of hemorrhage by deep learning," *Biomedical Signal Processing and Control*, vol. 71, p. 103085, 2022.

[52] U. Raghavendra, A. Gudigar, P. Kasula, Y. Chakole, A. Hegde, C. P. Ooi, E. J. Ciaccio, U. R. Acharya *et al.*, "Automated intracranial hematoma classification in traumatic brain injury (tbi) patients using meta-heuristic optimization techniques," in *Informatics*, vol. 9, no. 1. Multidisciplinary Digital Publishing Institute, 2022, p. 4.

[53] W. M. D. W. Zaki, M. F. A. Fauzi, R. Besar, and W. S. H. M. W. Ahmad, "Abnormalities detection in serial computed tomography brain images using multi-level segmentation approach," *Multimedia Tools and Applications*, vol. 54, pp. 321–340, 2011.

[54] M. Burduja, R. T. Ionescu, and N. Verga, "Accurate and efficient intracranial hemorrhage detection and subtype classification in 3d ct scans with convolutional and long short-term memory neural networks," *Sensors*, vol. 20, no. 19, p. 5611, 2020.

[55] T. Akilan, Q. J. Wu, and H. Zhang, "Effect of fusing features from multiple dcnn architectures in image classification," *IET Image Process.*, vol. 12, no. 7, pp. 1102–1110, 2018.

[56] J. He, "Automated detection of intracranial hemorrhage on head computed tomography with deep learning," in *Proceedings of the 2020 10th Intl. conf. biomedi. engineer. and technol.*, 2020, pp. 117–121.

[57] N. Farzaneh, C. A. Williamson, C. Jiang, A. Srinivasan, J. R. Bapuraj, J. Gryak, K. Najarian, and S. R. Soroushmehr, "Automated segmentation and severity analysis of subdural hematoma for patients with traumatic brain injuries," *Diagnostics*, vol. 10, no. 10, p. 773, 2020.

[58] I. Kumar, C. Bhatt, and K. U. Singh, "Entropy based automatic unsupervised brain intracranial hemorrhage segmentation using ct images," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2589–2600, 2022.

[59] H. Yao, C. Williamson, J. Gryak, and K. Najarian, "Brain hematoma segmentation using active learning and an active contour model," in *Bioinformatics and Biomedical Engineering: 7th International Work-Conference, IWBBIO 2019, Granada, Spain, May 8-10, 2019, Proceedings, Part II 7*. Springer, 2019, pp. 385–396.

[60] R. S. Barros, W. E. van der Steen, A. M. Boers, I. Zijlstra, R. van den Berg, W. El Youssoufi, A. Urwald, D. Verbaan, P. Vandertop, C. Majoie *et al.*, "Automated segmentation of subarachnoid hemorrhages with convolutional neural networks," *Informatics in Medicine Unlocked*, vol. 19, p. 100321, 2020.

[61] J. L. Wang, H. Farooq, H. Zhuang, and A. K. Ibrahim, "Segmentation of intracranial hemorrhage using semi-supervised multi-task attention-based u-net," *Applied Sciences*, vol. 10, no. 9, p. 3297, 2020.

[62] H. Yao, C. Williamson, R. Soroushmehr, J. Gryak, and K. Najarian, "Hematoma segmentation using dilated convolutional neural network," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 5902–5905.

[63] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[64] H. Yao, C. Williamson, J. Gryak, and K. Najarian, "Automated hematoma segmentation and outcome prediction for patients with traumatic brain injury," *Artificial Intelligence in Medicine*, vol. 107, p. 101910, 2020.

[65] J. Cho, K.-S. Park, M. Karki, E. Lee, S. Ko, J. K. Kim, D. Lee, J. Choe, J. Son, M. Kim *et al.*, "Improving sensitivity on identification and delineation of intracranial hemorrhage lesion using cascaded deep learning models," *Journal of digital imaging*, vol. 32, pp. 450–461, 2019.

[66] O. Eluyode and D. T. Akomolafe, "Comparative study of biological and artificial neural networks," *European Journal of Applied Engineering and Scientific Research*, vol. 2, no. 1, pp. 36–46, 2013.

[67] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[68] O. Caglayan, L. Barrault, and F. Bougares, "Multimodal attention for neural machine translation," *arXiv preprint arXiv:1609.03976*, 2016.

[69] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 452–461.

[70] R. Sahba, N. Ebadi, M. Jamshidi, and P. Rad, "Automatic text summarization using customizable fuzzy features and attention on the context and vocabulary," in *2018 world automation congress (WAC)*. IEEE, 2018, pp. 1–5.

[71] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *Intl. conf. machine learn.* PMLR, 2021, pp. 10 096–10 106.

[72] ——, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Intl. conf. machine learn.* PMLR, 2019, pp. 6105–6114.

[73] Z. Yao, A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. Mahoney, "Adahessian: An adaptive second order optimizer for machine learning," in *proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 10 665–10 673.

[74] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelli. and machine learn. for multi-domain operations applicat.*, vol. 11006. SPIE, 2019, pp. 369–386.

[75] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that?" *arXiv preprint arXiv:1611.07450*, 2016.

[76] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE Intl. conf. on computer vision*, 2017, pp. 618–626.

[77] H. Ko, H. Chung, H. Lee, and J. Lee, "Feasible study on intracranial hemorrhage detection and classification using a cnn-lstm network," in *42nd Annual Intl. conf. the IEEE Engineer. in Medicine & Biology Society*, 2020, pp. 1290–1293.

[78] Y. Guo, Y. He, J. Lyu, Z. Zhou, D. Yang, L. Ma, H.-t. Tan, C. Chen, W. Zhang, J. Hu *et al.*, "Deep learning with weak annotation from diagnosis reports for detection of multiple head disorders: a prospective, multicentre study," *The Lancet Digital Health*, vol. 4, no. 8, pp. e584–e593, 2022.

[79] S. Santhoshkumar, V. Varadarajan, S. Gavaskar, J. J. Amalraj, and A. Sumathi, "Machine learning model for intracranial hemorrhage diagnosis and classification," *Electronics*, vol. 10, no. 21, p. 2574, 2021.

[80] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

# Appendix

## Appendix A: IEEE Permission to Reprint

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Lakehead University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html and https://www.ieee.org/publications/rights/author-rights-responsibilities.html to learn how to obtain a License from RightsLink.

# Appendix B: Source Code

The source codes of this thesis are available at GitHub.