

BRAIN STROKE ANALYSIS FROM NON-CONTRAST BRAIN CT AND PATH-PLANNING FOR ROBOT-ASSISTED THROMBECTOMY

A PROJECT REPORT

Submitted By

A. ANIRUDH 195001015

KARTHIK D 195001047

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam - 603110

May 2023

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled **“BRAIN STROKE ANALYSIS FROM NON-CONTRAST BRAIN CT AND PATH-PLANNING FOR ROBOT-ASSISTED THROMBECTOMY”** is the *bonafide* work of **“A. Anirudh (195001015) and Karthik D (195001047)”** who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. T. T. Mirnalinee

Head of the Department

Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Dr. T. T. Mirnalinee

Head of the Department

Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Place:

Date:

Submitted for the examination held on.....

Internal Examiner

External Examiner

ACKNOWLEDGEMENTS

I thank GOD, the almighty for giving me strength and knowledge to do this project.

I would like to thank and deep sense of gratitude to my guide **Dr. T. T. MIRNALINEE**, Associate Professor, Department of Computer Science and Engineering, for his valuable advice and suggestions as well as his continued guidance, patience and support that helped me to shape and refine my work.

My sincere thanks to **Dr. T. T. MIRNALINEE**, Professor and Head of the Department of Computer Science and Engineering, for her words of advice and encouragement and I would like to thank our project Coordinator **Dr. B. BHARATHI**, Associate Professor, Department of Computer Science and Engineering for her valuable suggestions throughout this project.

I express my deep respect to the founder **Dr. SHIV NADAR**, Chairman, SSN Institutions. I also express my appreciation to our **Dr. V. E. ANNAMALAI**, Principal, for all the help he has rendered during this course of study.

I would like to extend my sincere thanks to all the teaching and non-teaching staffs of our department who have contributed directly and indirectly during the course of my project work. Finally, I would like to thank my parents and friends for their patience, cooperation and moral support throughout my life.

A. Anirudh

Karthik D

ABSTRACT

Brain stroke is among the most common reason for death around the world [25]. Brain imaging methods like magnetic resonance imaging (MRI), diffuse optical imaging, functional MRIs, and CT are quite helpful for initial screening. However, sophisticated imaging techniques like MRI require high operating cost and a well-trained operator. Often times, CT is the most convenient and quick imaging modality. Toward the diagnosis of strokes, physical and manual examination of the patient is performed to determine an appropriate course of treatment, aided by brain images. This process is made complex by the contrasting and divergent treatments required for the two kinds of stroke – ischemic and hemorrhagic – requiring accurate lesion type identification. Furthermore, an accurate segmentation of the infarct location is particularly useful for subsequent treatment. However, manually performing this diagnosis is extremely time-consuming for the already overburdened radiologist, and is prone to human-error. Thus, there is a need for an effective automated system for diagnosing strokes that can help physicians quickly start treatment after stroke onset. Hence, in this work, we proposed to develop a deep learning -based computation pipeline to process CT slices to detect stroke type and localize infarct location. Furthermore, we propose to reconstruct the CT slices into volumetric models while correcting errors in infarct localization, and potentially draw out diagnostic measurements that could help with subsequent treatment. Finally, we also propose a path planning algorithm to treat strokes using a six degree of freedom (DOF) arm. This Q-learning based method is simulated qualitatively on benchmark virtual environments.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
1 INTRODUCTION	1
1.1 MOTIVATION	1
2 PROBLEM STATEMENT AND OBJECTIVES	4
3 LITERATURE SURVEY	5
3.1 BRAIN STROKE SEGMENTATION	5
3.2 THROBECTOMY PATH PLANNING	10
4 PROPOSED SOLUTION ARCHITECTURE	11
4.1 SCAN PROCESSING MODULE	12
4.1.1 Data Collection	13
4.1.2 Slice-level Annotation to Establish Ground Truth	14
4.1.3 Data Pre-processing	14
4.1.4 Detection and Segmentation Model Building	16
4.1.5 Visualization of Predicted Patterns on the NCCT Slices . .	17
4.1.6 Volumetric Reconstruction and Correction	18
4.1.7 Diagnostic Inference from Volumes	18
4.1.8 Validation from Radiologists.	19

4.2	ROBOTIC PATH PLANNING MODULE	20
4.2.1	Distance Metrics	20
4.2.2	Simulated Annealing	22
5	PROOFS OF HYPOTHESES USED	23
5.1	ON THE EFFECTIVENESS OF SLICE-WISE SEGMENTATION	24
5.1.1	Statement	24
5.1.2	Explanation	24
5.1.3	Proof 1 – By Induction	25
5.1.4	Proof 2 – By Contradiction	30
5.2	ON THE FEASIBILITY OF DIMENSIONALITY REDUCTION FOR SEGMENTATION	31
5.2.1	Statement	31
5.2.2	Explanation	31
5.2.3	Proof	32
5.3	ON THE CONVERGENCE OF A NOVEL DISTANCE METRIC FOR PATH PLANNING IN ENVIRONMENTS WITH OBSTACLES	37
5.3.1	Statement	37
5.3.2	Explanation	37
5.3.3	Proof	38
6	EXPERIMENTS AND RESULTS	42
6.1	EXPLORATORY DATA ANALYSIS	42
6.1.1	Classification Models	43
6.1.2	Segmentation Models	43

6.1.3	Path Planning for Thrombectomy	45
7	CONCLUSION AND FUTURE WORK	48
7.1	CONCLUSION	48
7.2	FUTURE WORK	49
A	Q-LEARNING AND ROBOTICS TECHNIQUES	50
A.1	Q-LEARNING	50
A.2	INVERSE KINEMATICS	51
A.3	OPENAI GYM	53
A.4	ROBOTICS, VISION & CONTROL TOOLBOX	54

LIST OF TABLES

6.1	Comparison of classification approaches on open-access datasets. .	43
6.2	Comparison of segmentation approaches on open-access datasets.	44

LIST OF FIGURES

1.1	Samples of CT slices to depict the infarcts arising from <i>Ischemic</i> and <i>Hemorrhagic</i> strokes.	3
3.1	Samples of CT slices to depict <i>overlapping infarcts</i> – infarcts of multiple types at the same image region.	9
4.1	Architecture of the proposed solution.	12
4.2	Overview of the NCCT analysis workflow following steps 1 through 6.	13
4.3	An unsupervised approach to segment hemorrhagic infarcts from brain CTs.	15
4.4	Potential pre-processing methods to apply on the NCCT slices before passing them to the neural network and computational analysis pipeline to correct artifacts.	16
4.5	Path planning workflow based on Q-learning for the 6-DOF robotic arm.	21
6.1	Qualitative comparison of segmentation methods.	44
6.2	Proposed network architecture.	45
6.3	Intermediate states of the 6-DOF robotic arm when implementing the planned path in one of the six successful environments successfully planned by the proposed algorithm.	47
6.4	Comparison of number of steps to reach the target between the proposed distance metric and the Euclidean distance metric in three environments where the proposed distance metric outperforms the Euclidean distance metric.	47

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

After heart disease, brain stroke is the most common reason for death around the world [26]. A majority of survivors need to live with changeless or long-term injury. Brain imaging methods like magnetic resonance imaging (MRI) and CT are quite helpful for a doctor in order to start the initial screening of the patient. There are also many imaging modalities for the analysis of brain, which may include X-ray imaging, diffuse optical imaging, magnetoencephalography, functional MRI and positron emission tomography [5, 45]. However, all these imaging techniques require high operating cost and well-trained operator, hence most of these imaging methods may not be available in all the clinics and hospitals.

Image classification is widely used in medical imaging [32]. However, for better accuracy of the classification system, results should be close to the manual diagnosis. Nowadays, deep learning has been extensively used as a classification method because it automatically calculates features within the convolutional layers of the deep system [1, 27]. The main advantage of using deep learning is that it outperforms other conventional methods for image classification [38]. Many deep learning methods have come into existence such as recurrent neural networks, long short-term memory (LSTM) [15], CNNs [30], deep belief nets (DBN) [14], etc. Among these methods, CNN has been generally utilized in computer vision and medical image processing problems like ImageNet, face

recognition, house numbers digit classification, patch classification from medical images etc. Classification methods other than deep learning methods are random forest (RF), -nearest neighbors (NN), decision tree (DT), multilayer perceptron (MLP), support vector machine (SVM) and many more [37, 39, 41].

In this research, an automatic classification method has been proposed to predict the category to which brain CT scan image belongs, along with the locations of the stroke infarcts for hemorrhagic and ischemic stroke types, and normal tissue. *Hemorrhagic stroke* occurs due to fragile blood vessel which burst and drains into the neighboring brain tissues. On the other side, *ischemic stroke* occurs when blood supply in the brain stops due to the presence of blood clots. Brain hemorrhage can also occur after ischemic stroke which results in a serious complication [22]. Sample images of the two types of stroke on CT slices are depicted in Figure ??, with infarct regions demarcated in red.

Towards the diagnosis of stroke, doctors first confirm symptoms from the patient or family members. The most important thing which will help in accurate identification is the stroke history. This is usually followed by neuroimaging of the patient. The most commonly used neuroimaging method used by experts is CT scan for diagnosing brain strokes, on account of their greater affordability. The prediction of stroke from CT scan images serve as the initial step towards the proper diagnosis of a patient. These images are then sent to a cardiovascular radiologist to identify the stroke type. Thereafter, physical and manual examination of the patient is performed to determine an appropriate course of treatment. However, this manual diagnosis procedure is extremely time-consuming for the already overburdened radiologist, and is prone to human-error. Many researchers have worked in this area for providing a

computer-aided diagnosis (CAD) system, in spite of the significant requirement, there is no clinically accepted CAD system for stroke [28, 31]. The system proposed by [28] was not automated and requires significant performance improvements to warrant automation. Peixoto and Filho [31] have worked on the classification of hemorrhagic and ischemic stroke CT scan images. Their method was developed for small dataset where each case had 100 images and for future work they have to test their method on large datasets. Thus, there is a need for an effective automated system for diagnosing strokes that can help physicians quickly start treatment after stroke onset. Hence, in this work, we proposed to develop a deep learning -based computation pipeline to process CT slices to detect stroke type and localize infarct location. Furthermore, we propose to reconstruct the CT slices into volumetric models while correcting errors in infarct localization, and cast the volume into a computationally pliable 3D model that can be used to draw out diagnostic measurements as well as for treatment simulations.

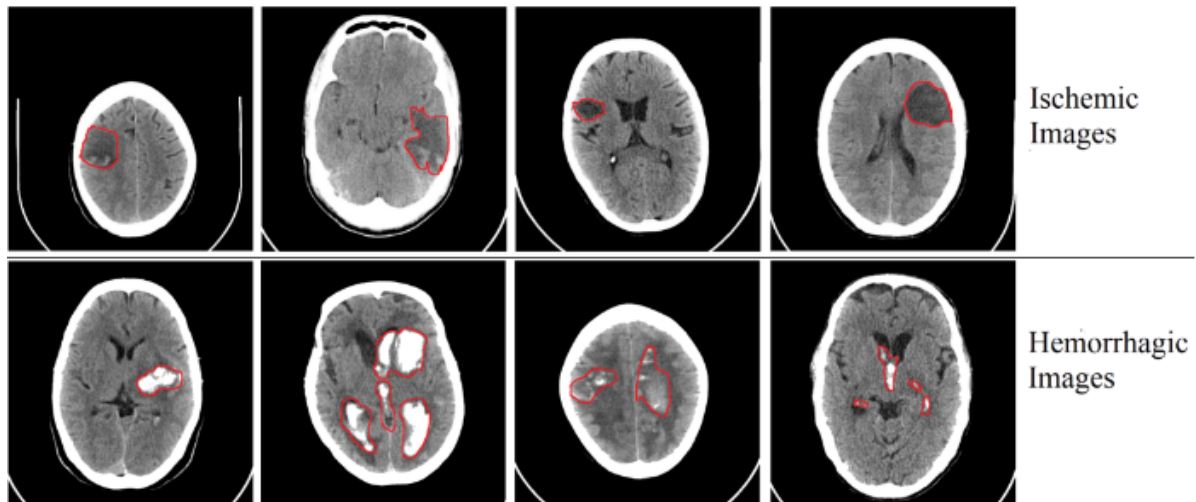


FIGURE 1.1: Samples of CT slices to depict the infarcts arising from *Ischemic* and *Hemorrhagic* strokes.

CHAPTER 2

PROBLEM STATEMENT AND OBJECTIVES

This research proposes to achieve the following objectives through data-driven computational modeling and analysis. The approach and methods that will be adopted in meeting these objectives is detailed in the subsequent section. The specific objectives that we sought to achieve in this project are as follows,

- To detect the presence of and differentiate between ischemic and hemorrhagic infarcts from non-contrast CT (NCCT) slices of the brain.
- To localize and segment the regions representing normal, ischemic infarcts, and hemorrhagic infarcts from the NCCT slice images.
- To reconstruct the volumetric NCCT image of the brain from the processed slices while simultaneously extrapolating and correcting infarct segmentation demarcations based on geometric constraints.
- To infer location information and other quantitative diagnostic measures from the reconstructed volumes to assess the condition such as cerebral infarct volume (CIT).
- To fuse patient-specific details into a computational model that will integrate the quantitative measures and parameters describing the infarct location with a generalized structural model of the brain.
- To devise a suitable path-planning strategy to perform autonomous thrombectomy using a robotic arm in the simulation environment with support for manual intervention in case of emergencies.

CHAPTER 3

LITERATURE SURVEY

3.1 BRAIN STROKE SEGMENTATION

In general, processing and analysis of medical images using computer-aided diagnosis (CAD) systems can help physicians in early identification of diseases [12, 17, 28, 43]. Herein, the proposal briefly describe some important image preprocessing and classification methods used so far in order to analyze brain strokes from medical images.

Saatman et al. [33] have discussed various classification techniques for traumatic brain injury so that proper treatment can be provided for that targeted injury. They discussed classification based on the severity of brain injury where various parameters which include Glasgow Coma Scale (GCS) have been used to identify the severity, however these procedures require many clinical trials. Other techniques which they have discussed are pathoanatomic classification, classification by physical mechanism, pathophysiology and prognostic modeling. According to them, for avoiding several clinical trails of patients new tools can be developed like dataset development, data sharing, data mining, and bioinformatics, etc. In 2008, Mozqueda et al. [40] have tested the hypothesis on clinical data of patients that acute ischemic stroke classification from CT or MR angiography images into major and minor stroke can be done by using Boston Acute Stroke Imaging Scale (BASIS) instrument. They have proved that BASIS

classification instrument is better than Alberta Stroke Program Early CT Score (ASPECTS) in predicting stroke.

Chawla et al. [2] presented an automatic method for the detection and classification of an abnormality from low-contrast CT images into three categories — hemorrhage, chronic and acute infarct. Their proposed technique comprises of three steps: enhancement of the image, recognition of mid-line symmetry and classification of abnormal slices. The main limitation of their work is that they have just shown results on the dataset and have not compared their results with any of the previous state-of-the-art methods by generating their results on their dataset. The accuracy obtained by them on their dataset was also not given and have just shown precision and recall values for the experiments. Griffis et al. [11] have also worked on ischemic stroke lesions identification where they utilized the T1-weighted MRI scans and predict the stroke region using Gaussian naïve Bayes classification method. Limitations of their method was that they were unable to detect very subtle WM lesions with very small extents. Shahangian and Pourghassem [35] proposed a method to segment and classify hemorrhage, where segmentation was based on a modified version of distance regularized level set evolution (MDRLSE). Thereafter, SVM classifier was used to classify images by extracting weighted grayscale histogram features where they achieved an accuracy of 94.13%. However, the complete dataset information is not given like under what parameters images have been acquired and the details of CT scanner. The other limitation is that, the number of images which they have used for classification were also very less for 5 different categories i.e. only 627 images. The segmentation of hemorrhagic stroke from CT scan images was proposed by Gautam et al. [9] using variant of fuzzy clustering named as modified robust fuzzy c-means clustering (MRFCM) and

used it with distance regularization level set evolution (DRLSE) method for the segmentation. However, their method was not able to identify very small lesions from the images. They have also proposed the method for the classification of a brain CT scan images into hemorrhagic, ischemic and normal [8]. A new feature descriptor known as the local gradient of gradient pattern (LG2P) descriptor has been proposed by them. The highest classification accuracy achieved by them on 900 image dataset is 83.11% and 86.11% using fine NN and cubic SVM. However, this accuracy is very less and for better results it must be improved. Incremental gradient descent decision boundary feature extraction method was proposed by Woo et al. [42] for the classification of UCI dataset where they have shown results on Parkinson speech, chess, molecular biology, gas sensor, and array drift datasets of UCI. Recently, some researchers have published a literature survey on feature extraction methods and they can be found in [3, 34], so readers are referred to these papers for more methods.

Subudhi et al. [37] have proposed a new method for the segmentation and classification of brain stroke from MR images where they used expectation–maximization and random forest classifier. The accuracy achieved by them was 93.4% on the dataset of 192 brain images. The limitation of this paper is that they do not have any discussion of the dataset like from where they collected it and how. Ortiz-Ramón et al. [29] also worked on MR images where they proposed a texture analysis method to identify whether the patient has a stroke or not and also identify whether that stroke is lacunar or cortical. In this paper, they have used various methods proposed by different researchers in order to carry out their work and references for them can be found in [37]. According to them the limitation of their work is inability to combine stroke and aging (old stroke lesion) datasets for the analysis of images showing recent cortical or

lacunar strokes. However, image normalization techniques can solve this issue by improving the image qualities.

Apart from the conventional machine learning methods which have been used very widely for the identification of diseases from the medical images, deep learning approaches are gaining attention because of the accurate diagnosis results. In 2012, Krizhevsky et al. proposed AlexNet as the CNN architecture and won the most difficult ImageNet large-scale visual recognition challenge (ILSVRC) [19]. Thereafter, it emerged as the revolution in the field of computer vision and machine learning. AlexNet has 5 convolutional and 3 fully connected layers where 2 dropout layers have been used with 0.5 probability. The best classification prediction with it can be achieved when the softmax layer is fed with the result of the last fully connected layer. In this architecture, the input image size should be $224 \times 224 \times 3$. However, in order to improve the accuracy many researchers have tried to develop new architectures. In 2014, VGG16 has come into existence, proposed by Simonyan and Zisserman [36]. The main contribution of their network is that they have made a very deep CNN architecture for the recognition of very large image dataset by increasing the network depth to 16–19 layers with 3×3 convolution filter. VGG16 consists of five batches of 13–16 convolutional layers and 3 fully connected layers. In the first layer the number of filters starts from 64 and increases by a factor of 2 for each layer after the max-pooling till it becomes 512. The input image size for this architecture remains the same as that of AlexNet i.e $224 \times 224 \times 3$. The other CNN architecture ResNet50 proposed by He et al. [13] is a very deep residual network and it won the first place in ILSVRC 2015. It is a 50 layer architecture instead of 16–19 layers, which was trained on the 1.28 million training images. The input image size for this architecture is $224 \times 224 \times 3$. After that, many variants of

CNN architectures have come into existence in order to provide better results for the diagnosis of the patients. Recently, deep learning has shown good classification accuracy on both training and testing datasets of breast cancer histopathology images [12]. The other work on the detection and classification of breast cancer makes use of four fully convolutional networks (FCN) [10]. CNN has also been used to detect colorectal cancer from the histology images [44]. Gao et al. [7] used deep CNNs to classify interstitial lung diseases (ILD) from CT images by using the entire image as a holistic input.

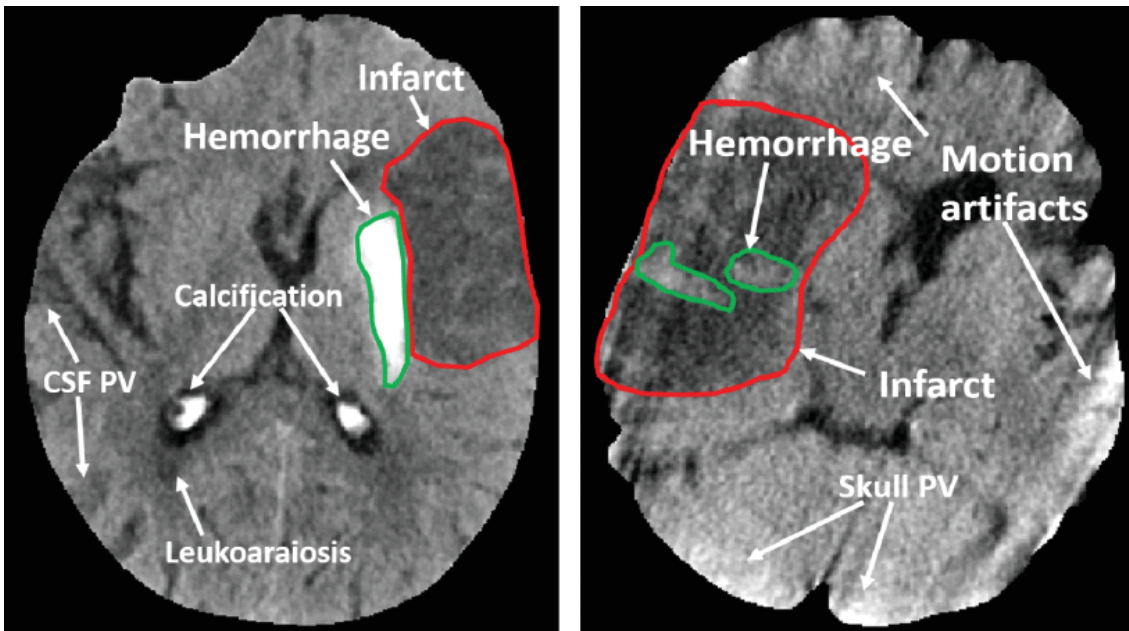


FIGURE 3.1: Samples of CT slices to depict *overlapping infarcts* – infarcts of multiple types at the same image region.

However, the work on the classification and localization of origin of brain strokes from CT/MR images still requires improvements for clinical applicability. Furthermore, the presence of overlapping infarcts of the two types – ischemic and hemorrhagic – on the CT slides, as depicted in Figure 3.1 gives rise to a *multi-label* segmentation problem, where each image pixel may belong to more than one class. Therefore, in this research, we propose to develop a new deep learning -based processing pipeline to classify brain stroke from CT scan images,

and further localize the infarct through volumetric reconstruction. Moreover, to the best of the authors' knowledge, this work is the first attempt to implement simultaneous volumetric reconstruction and correction of segmentation to improve demarcation performance and evolve a pliable computational model that can be used to infer quantitative diagnostic measures and with further extensions, for simulating surgical procedures as well.

3.2 THROBECTOMY PATH PLANNING

For thrombectomy path planning, Jia Y. et al. [18] proposes a learning-based obstacle avoidance method using Q-learning for path planning in six-axis robotic arms using reinforcement learning for autonomous obstacle avoidance. The strategy prioritizes planning the obstacle avoidance path for the terminal point of the mechanical arm and then uses the calculated terminal path to plan the poses of the mechanical arm. If the mechanical arm cannot avoid obstacles within the limit of the safe distance for points on the terminal path, this strategy records those points as new obstacles and plans a new obstacle avoidance path for the terminal of the mechanical arm. This process is looped until the correct path is calculated. This method was adopted as a baseline with noteworthy modifications in our work.

CHAPTER 4

PROPOSED SOLUTION ARCHITECTURE

In recent years, research in deep learning has demonstrated successful application of deep learning and convolutional neural networks for image classification, including medical image analysis. In this study, we propose to develop a deep neural network -based processing pipeline for classifying different stroke patterns on NCCT images, and localizing the infarct for downstream treatment formulation.

Toward solving the problem, we will collect NCCT imaging slides of study patients. Following this, the digital images will be randomly partitioned into two sets, one for model training and another as an independent test set for the final evaluation of the model. Care will be taken to avoid overlapping slides from a given patient across the training and test sets. Each slice will be manually assessed and its regions will be labeled as either *normal* (0), *ischemic infarction* (1), *hemorrhagic infarction* (2) by a trained cardiovascular radiologist. Thus, labeled slice image will form the input for training the deep learning -based computational pipeline. The model will be developed to take in square patches or tiles as inputs and output a prediction probability for each of the three classes: 0, 1, and 2 in a multi-label setting. Different model parameters including network weights, test for multiple network layers, learning rates, loss functions will be considered to fine-tune the model and formulate a model with optimal parameter settings. An unseen held-out subset of the slices data will be used for independently validating and evaluating performance and accuracy of the trained model. Finally, the models' ability to recognize the individual segmentation

classes will be analyzed. As a final step, the processed image slides with infarct region annotations, as predicted by the developed computational model, will be used to perform volumetric reconstruction and infarct region correction. Furthermore, we will visualize the model's predictions overlaying color-coded dots on patches for which the model predicted patterns. This will aid in quickly identifying regions of the slice(s) from a given patient containing abnormal tissue. An overview of the proposed solution architecture is presented in Figure 4.1.

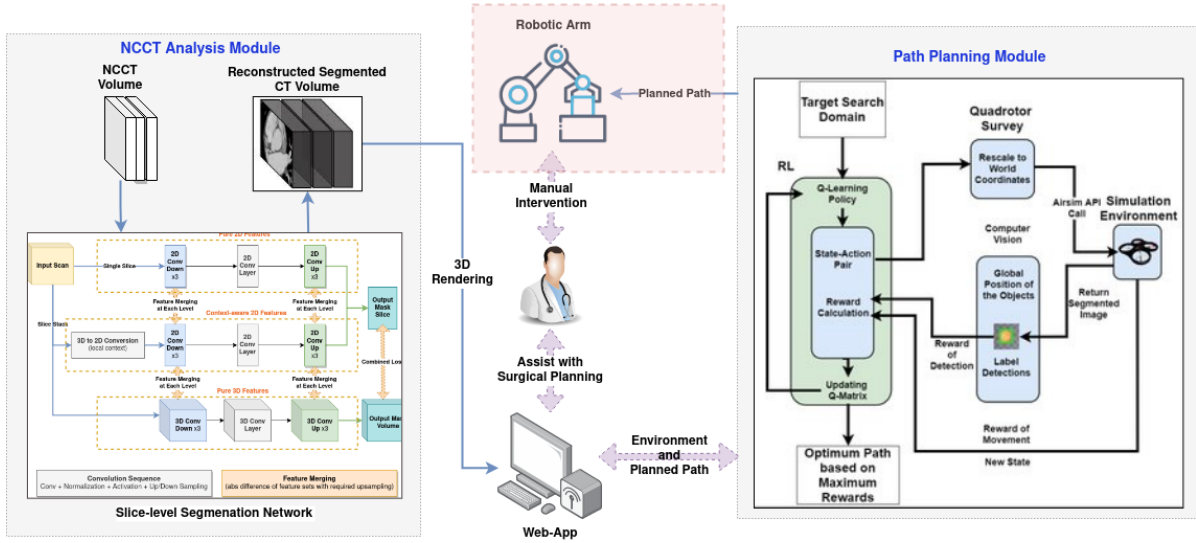


FIGURE 4.1: Architecture of the proposed solution.

4.1 SCAN PROCESSING MODULE

An overview of the NCCT scan processing module is presented in Figure 4.2.

4.1.1 Data Collection

To develop and evaluate our computational model for detecting distinctive infarct patterns in CTs, we will acquire volumetric NCCT slices in a digital image format, typically DICOM or NIfTI. Following this, the images will be randomly partitioned into two sets: (a) one for *model building* – i.e., training and validation (about three-fourths of the dataset), and (b) for *model evaluation* (about one-fourth of the dataset).

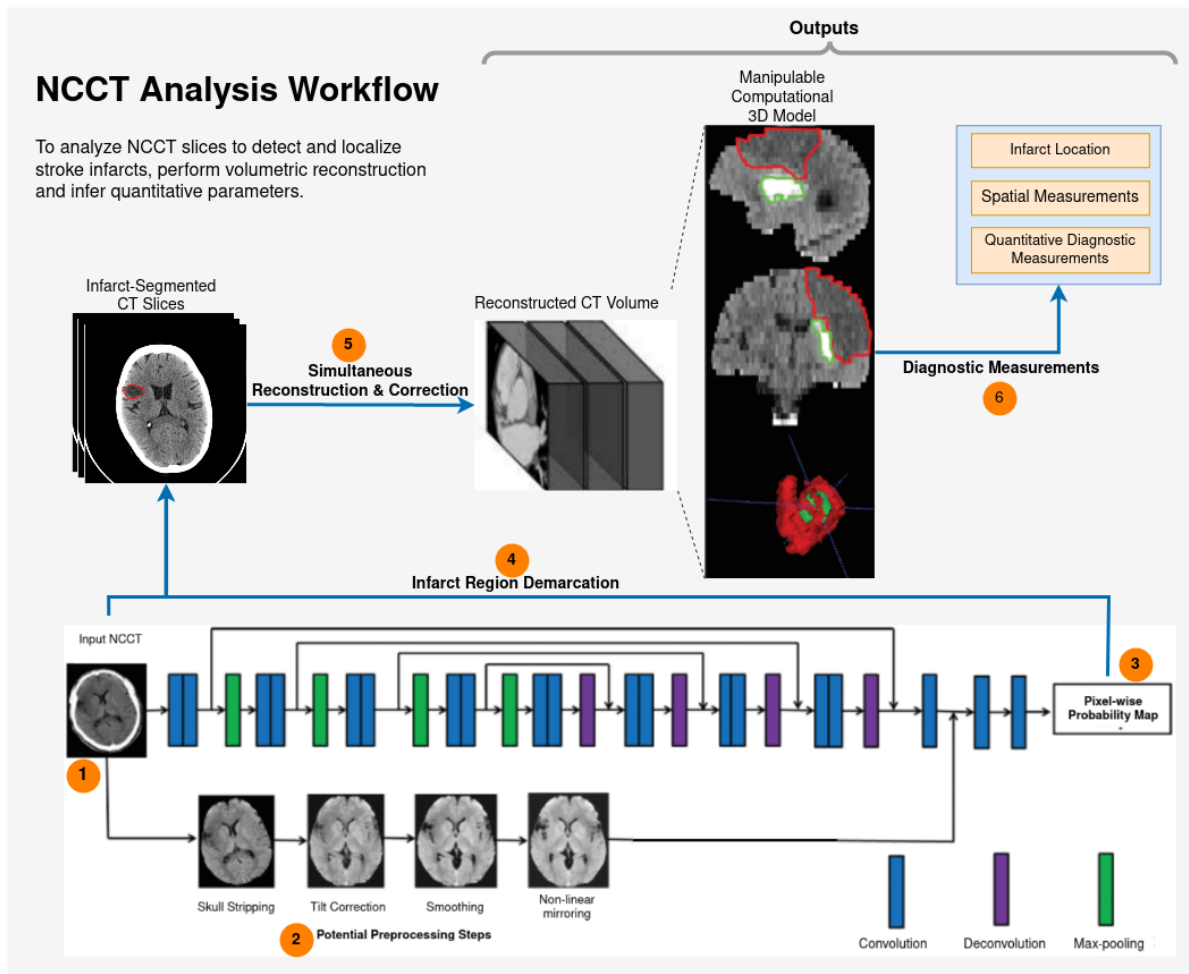


FIGURE 4.2: Overview of the NCCT analysis workflow following steps 1 through 6.

4.1.2 Slice-level Annotation to Establish Ground Truth

All NCCT slice images will be manually labeled by one or more medical experts. Each image corresponding to the model building dataset will be annotated with scan regions as either one of the two patterns or normal/non-applicable. These are: *normal (0)*, *ischemic infarction (1)*, *hemorrhagic infarction (2)*. Ideally, we would resort to multiple independent annotation results by the experts and correct for inter-rater biases to ensure high quality of the ground truth. For the purpose of annotations, a granulated user-friendly tool will be provided to the medical expert(s) to perform manual annotation at the super-pixel image region level, with the ability to select large regions of the image at a time and label them. Furthermore, an unsupervised clustering-based segmentation algorithm will be applied to facilitate easy manual annotation of infarct regions on the NCCT slices. This algorithm is presented in Figure 4.3. This will save time and effort on their part. In addition, case-specific comments will be collected for downstream evaluation and analysis of the developed processing framework. The test set will be used, once the model development is complete, to evaluate its performance and compare it against that of expert annotators.

4.1.3 Data Pre-processing

The input data in the form of CT slices will be processed by a computation processing pipeline that comprise deep learning network to analyze images and draw inferences. Initially, basic pre-processing methods to ensure consistency of image quality and exposure statistics across the dataset will adopted, and further pre-processing workflows may be integrated at a later stage during model

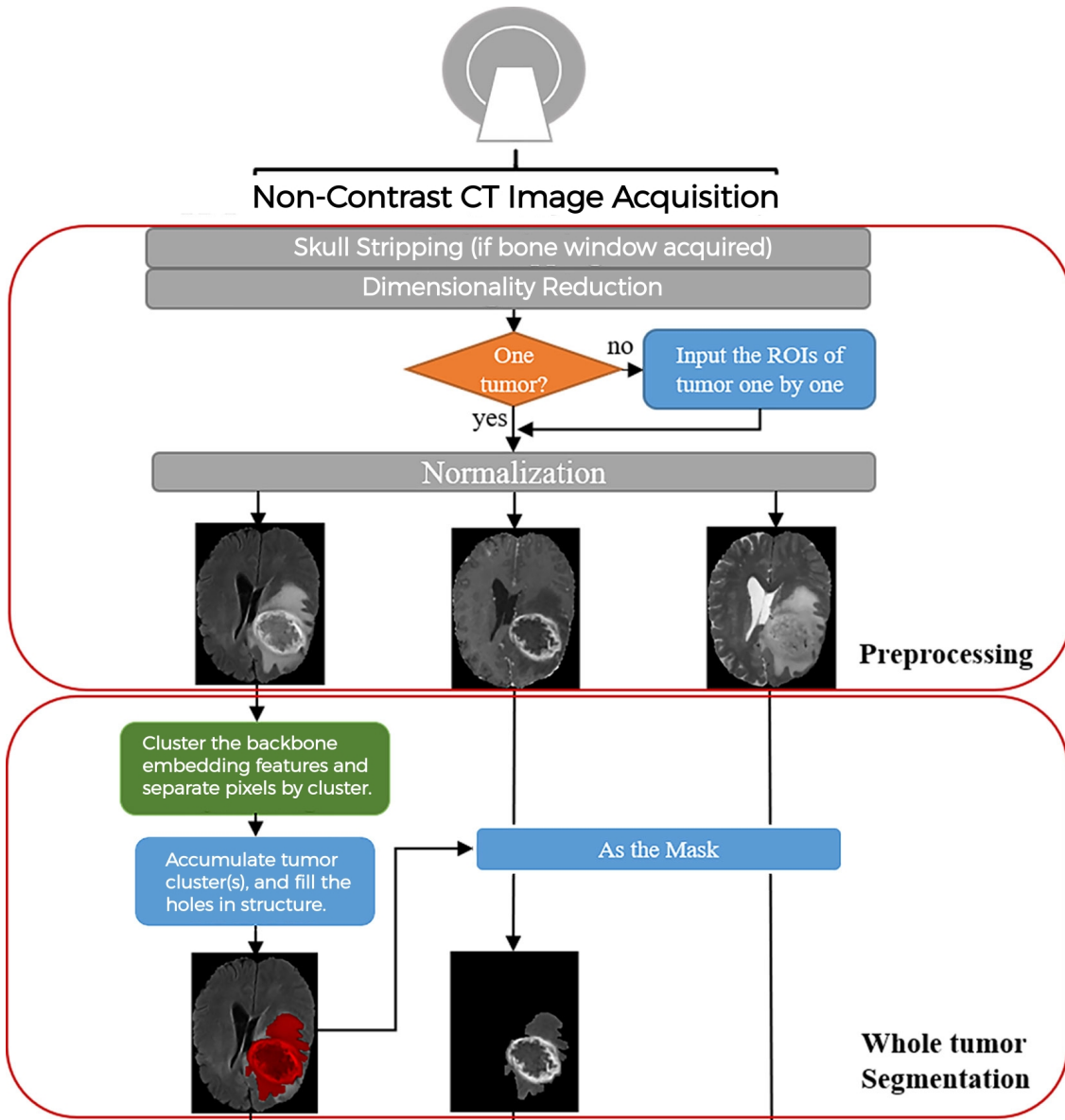


FIGURE 4.3: An unsupervised approach to segment hemorrhagic infarcts from brain CTs.

building by empirically determining the specific image artifacts to be managed and the method required to do so. Some of the basic methods include: automatic resizing of training images to contain a fixed dimension of pixels; apply color jittering on the brightness, contrast, saturation, and hue of each image to neutralize the differences across the dataset; and augmentation of image data to randomly flip and rotate them along the horizontal and vertical axes to obtain variability in the training data, and to ensure robust model development, among

others. Some more potential pre-processing methods that may be employed, should the need arise, are depicted in Figure 4.4.

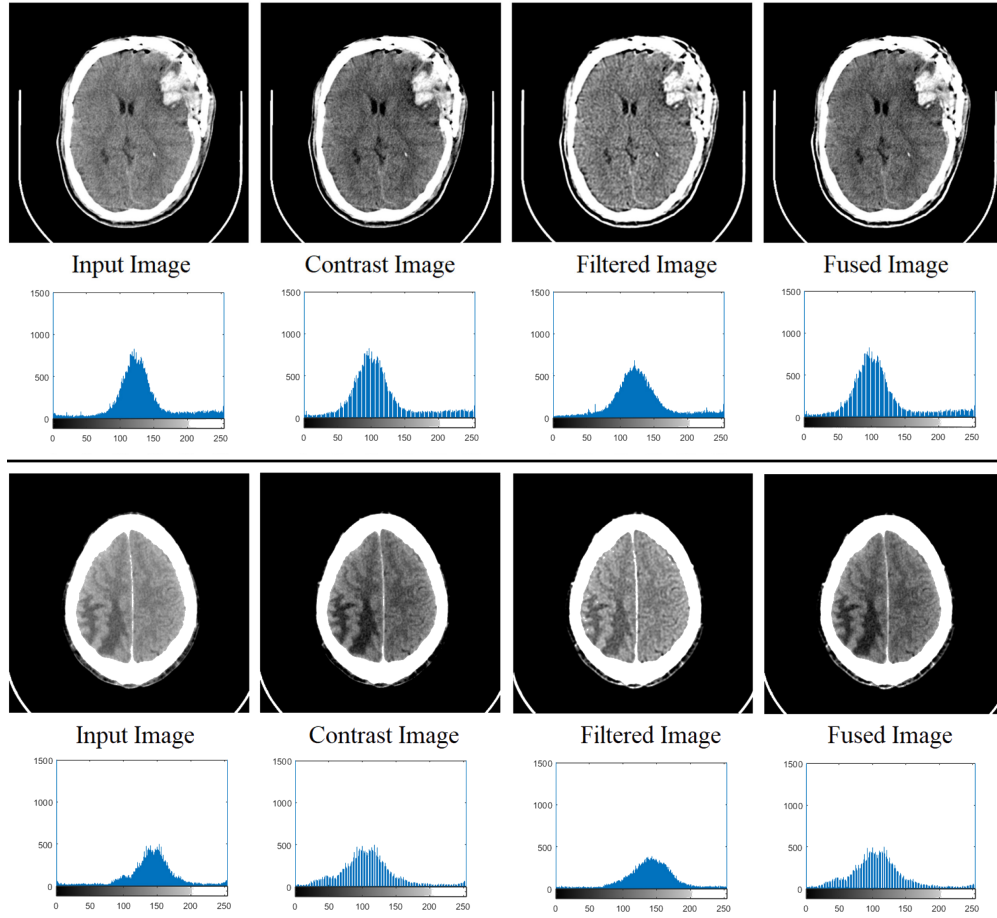


FIGURE 4.4: Potential pre-processing methods to apply on the NCCT slices before passing them to the neural network and computational analysis pipeline to correct artifacts.

4.1.4 Detection and Segmentation Model Building

Before training the classification and segmentation models, 90% of the model building data will be randomly selected for training and 10% for internal validation. The models will be implemented using the deep learning library Torch available for Python. Torch is a high-level neural networks API, capable of running on top of various backend architectures. The models will be trained

under a supervised setting, with NCCT slices fed as input, and outputs corrected against their corresponding expert-annotated and inter-rater corrected NCCT slices. Different tests will be conducted to optimize for the number of epochs, initial learning rates as well as the optimal depth of the neural network and other functionally configurable portions of the computational pipeline. The output of the models will be configured to output probability vectors p per stroke type, normalized to 1. Finally, the learning curves (accuracy vs. no. of iterations) for both training and testing will be plotted to determine if there is a generalization of the deep learning network model(s) to the unseen validation/test data i.e., no over-fitting has occurred. Two state-of-the-art slice-based image segmentation neural networks, namely Slice-Aware Net (SA-Net) [25] and M-Net [24] were adopted as improved models. In addition, a combination of the two models was also implemented to combine the strengths of these state-of-the-art approaches – long-range dependency capturing ability of SA-Net and spatial context capturing ability of M-Net.

4.1.5 Visualization of Predicted Patterns on the NCCT Slices

The processed and region-predicted NCCT slices will be visualized by overlaying color-coded dots on patches for which our model predicted a stroke infarct pattern, or each type of the stroke. This visualization will allow us to the decisions generated by the model and further aid to gain insights into the model's classification method.

4.1.6 Volumetric Reconstruction and Correction

With the infarct segmentation workflows in place, the subsequent steps involve reconstructing the volumetric CT of the patient’s brain with annotated infarct regions, labeled by type – ischemic and hemorrhagic. This process will involve stacking the processed NCCT slides whilst simultaneously correcting the neighbors for potential errors in region segmentation. These corrections will be automated based on geometric, spatial, and shape-based constraints of adjacent slides such as contour evolution methods [20], depicted in Figure ???. The correction strategy will not only improve the overall localization accuracy of the ischemic strokes facilitating subsequent use of the reconstructed volume, but will save compute time due to simultaneous reconstruction and correction and improve model inference through a more interpretable basis for reconstruction. Ultimately, this step yields an annotated volumetric model of the brain region that is pliable for exploratory analysis on a computer.

4.1.7 Diagnostic Inference from Volumes

The ultimate goal is to localize infarct types and locations in the spatial context. In addition, we also propose to estimate quantitative diagnostic measures. The annotated volume of the patient’s brain can be easily exploited to draw out any form of geometric, spatial, or structural measurements, and even a combination thereof. Furthermore, the presence of each type of infarct can be individually quantified to present summary inferences of the patient diagnosis to help clinicians devise an ideal treatment strategy.

4.1.8 Validation from Radiologists.

At the completion of the complete processing pipeline, two sets of validations will be performed to ensure correctness of the overall computational pipeline – (1) *the volume-level annotations* will be validated by radiologists, and (2) *the slice-level annotations* of the held-out test-set will be compared to the model predictions. *Evaluation 1* will be qualitative in nature, owing to the tediousness of manually annotating volumetric regions. However, this evaluation will be supported by *evaluation 2*. In conjunction, the two sets of validations will provide sufficient evidence to verify the complete proposed computational pipeline. To evaluate the slide-level performance, the model’s labels will be compared against those of the radiologists’ by calculating an inter-rater reliability metric called Cohen’s kappa score [23], as it has been adapted as a standard metric when it comes to annotation of medical images. Thus, between every two sets of annotations from radiologists and/or the computational pipeline, both pd scores and kappa score per class will be computed. K_{pd} refers to the percentage of NCCT slices in which two annotators agreed on the same annotation. Kappa scores per infarct class, on the other hand, estimates the detection of a given infarct pattern or type between two sets of annotations. The annotations will be quantified using a suitable metric as deemed suitable during discussions with the experts.

4.2 ROBOTIC PATH PLANNING MODULE

Building atop the proposed Q-learning strategy in [18], and deriving from methods for obstacle avoidance in robotics, such as the artificial potential field-based industrial robot station conversion obstacle avoidance system, genetic algorithms, and RBF neural networks have limitations in complex environments and high-dimensional multi-constraint path planning problems. The approach prioritizes planning the obstacle avoidance path for the terminal point of the mechanical arm and then uses the calculated terminal path to plan the poses of the mechanical arm. If the mechanical arm cannot avoid obstacles within the limit of the safe distance for points on the terminal path, the strategy records those points as new obstacles and plans a new obstacle avoidance path for the terminal of the mechanical arm, looping this process until the correct path is calculated.

An overview of the path planning workflow is presented in Figure 4.5. The proposed method makes use of a novel distance function and simulated annealing techniques for exploratory actions.

4.2.1 Distance Metrics

The distance is calculated using a custom metric that incorporates the distance between the current and previous arm positions and the target point. The function uses vector algebra to determine the shortest distance between the line segment and the third point. Given two vectors \mathbf{u} and \mathbf{v} with n elements, the maximum distance between them using the Chebyshev and Minkowski distance metrics, with

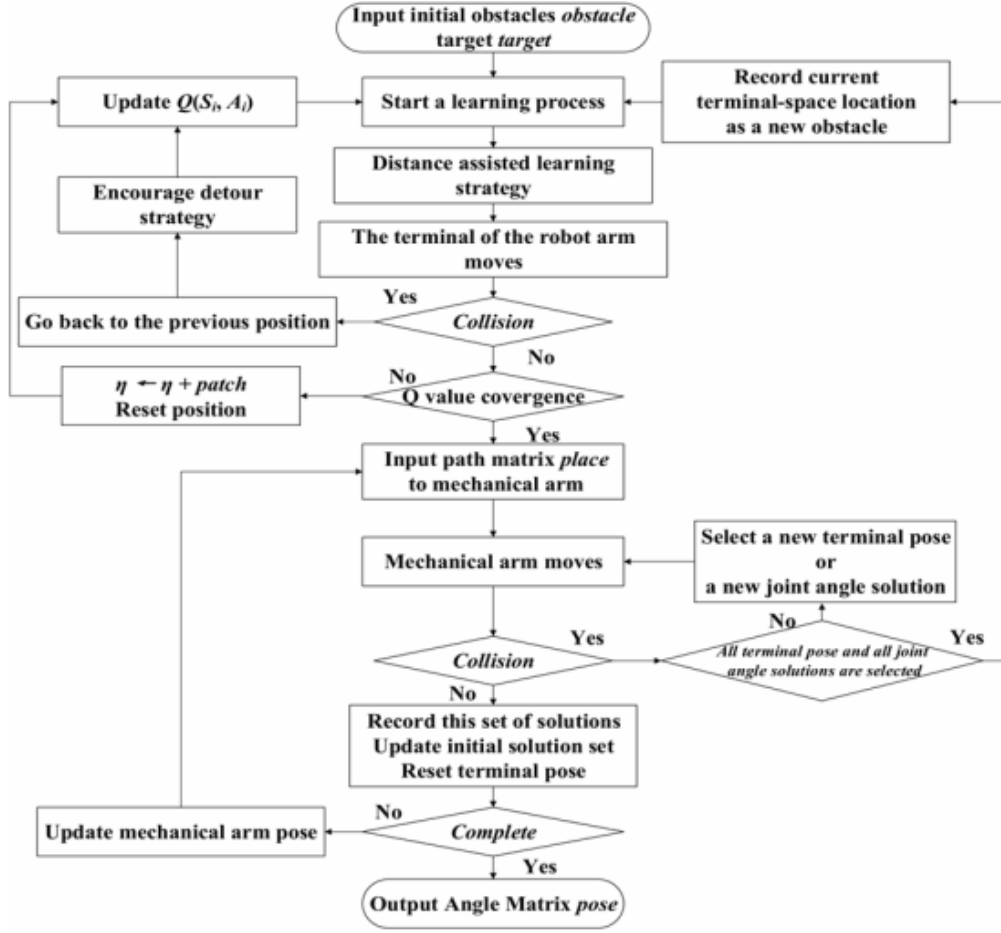


FIGURE 4.5: Path planning workflow based on Q-learning for the 6-DOF robotic arm.

p defined as the difference between the Euclidean distances of the current arm position and previous arm position to the target point, described in Equation 4.1.

$$\max \left(\max_{i=1}^n |u_i - v_i|, \left(\sum_{i=1}^n |u_i - v_i|^p \right)^{1/p} - \left(\sum_{i=1}^n |u'_i - v'_i|^p \right)^{1/p} \right) \quad (4.1)$$

where \mathbf{u}' and \mathbf{v}' are the previous arm position and target, respectively, $\|\cdot\|_2$ denotes the Euclidean norm. Further, p is computed using Equation 4.2.

$$\|\text{current arm position} - \text{target}\|^2 - \|\text{previous arm position} - \text{target}\|^2 \quad (4.2)$$

4.2.2 Simulated Annealing

A temperature-based simulated annealing strategy is used to decide between greedy (maximizing reward) and exploratory (random) actions.

Let $E(s)$ be the energy (or cost) associated with state s , ΔE be the energy difference between the current state and the proposed next state, and $T(t)$ be the temperature at iteration t . The probability of accepting a worse state at temperature $T(t)$ is given by the Boltzmann distribution:

$$P(\Delta E, T(t)) = \exp\left(-\frac{\Delta E}{T(t)}\right) \quad (4.3)$$

To decide whether to take a greedy or exploratory action, start with a high initial temperature T_0 and gradually decrease it over time according to a cooling schedule, such as:

$$T(t) = T_0 \cdot \alpha^t \quad (4.4)$$

where α is the cooling rate, t is the iteration number, and $T(t)$ decreases exponentially as t increases. At each iteration, we can generate a random neighboring state of the current state, and compute the energy difference ΔE . If $\Delta E \leq 0$, we accept the new state as the current state. Otherwise, we accept the new state with probability $P(\Delta E, T(t))$.

CHAPTER 5

PROOFS OF HYPOTHESES USED

Portions of the adopted methodology in the proposed solution was designed based on three hypotheses. First, a slice-wise segmentation approach was used amidst several other approaches in literature owing to its advantages in model inference time in the context of deep learning models and spatial context awareness, among others. Second, to facilitate easy manual annotation of NCCT volumes by experts, an approximate unsupervised segmentation algorithm was designed. A primitive proof of the feasibility and applicability of these methods is presented in this section. we employ a novel distance metric for the purpose of path planning of a robotic arm in environments containing obstacles. Our proof shows that this distance metric exhibits faster convergence rates as compared to the commonly used Euclidean distance metric. Finally, we employ a novel distance metric for the purpose of path planning of a robotic arm in environments containing obstacles. Our proof shows that this distance metric exhibits faster convergence rates as compared to the commonly used Euclidean distance metric.

5.1 ON THE EFFECTIVENESS OF SLICE-WISE SEGMENTATION

5.1.1 Statement

Slice-wise segmentation is at least as effective as global volumetric segmentation of the CT slices to identify infarct regions.

5.1.2 Explanation

We begin by noting that many structures in medical imaging are naturally represented as a stack of 2D images, with the relevant features of the structure spread out across multiple slices. For example, the edges of the lungs are often visible on multiple slices, and may be difficult to identify on any individual slice. By examining the entire stack of slices, we can identify the edges more easily and accurately. Therefore, slice-by-slice segmentation can be an effective way to segment these structures.

To make this more concrete, let's consider a simplified example of a 3D CT image of a cube. Suppose we wish to segment the cube from the background. We could apply a thresholding algorithm to the entire 3D volume at once, but this may not be effective if the cube is not easily distinguishable from the background. Instead, we could use slice-by-slice segmentation. We begin by segmenting the first slice using a thresholding algorithm. We then move on to the second slice, and use the segmentation of the first slice as a prior to guide the segmentation of the second slice. Specifically, we can use the segmentation of the first slice to

identify a range of intensity values that are likely to correspond to the cube, and then apply a threshold within this range to the second slice. We repeat this process for each subsequent slice, using the previous segmentations as priors to guide the segmentation of the current slice.

The key advantage of this approach is that it can be more robust to noise and other artifacts that may be present in the image. If we were to apply a global thresholding algorithm to the entire 3D volume at once, we may include or exclude voxels that should be part of the cube due to variations in the intensity values across the volume. However, by segmenting each slice individually and using the previous segmentations as priors, we can incorporate more information about the structure into the segmentation process, and potentially achieve better results.

Therefore, we conclude that slice-by-slice segmentation can be an effective way to segment 3D CT images, particularly for structures that are naturally represented as a stack of 2D images. By segmenting each slice individually and using the previous segmentations as priors, we can incorporate more information about the structure into the segmentation process and potentially achieve better results than by applying a global thresholding algorithm to the entire 3D volume at once.

5.1.3 Proof 1 – By Induction

Assumptions

1. The CT scan is of a single anatomical structure, such as a lung, liver, or kidney.

2. The structure is contiguous, meaning that there are no gaps or holes in the 3D volume.
3. The structure has a consistent appearance across all slices, meaning that it looks the same in each 2D image.

Definitions

1. A voxel is a 3D pixel in the CT image, with a certain intensity value.
2. A segmentation is a labeling of voxels as belonging to either the structure of interest or the background.

Let $I(x, y, z)$ be a 3D CT image, where x, y, z denote the spatial coordinates. Our goal is to segment a target structure S from the background. We assume that S is connected and has a reasonably smooth boundary in the image.

Suppose we apply a global thresholding algorithm to the entire 3D volume at once, using a threshold value T . Let S_T be the resulting segmentation, defined as:

$$S_T(x, y, z) = \begin{cases} 1 & \text{if } I(x, y, z) > T \\ 0 & \text{otherwise} \end{cases}$$

Note that S_T is a binary image that indicates which voxels belong to the target structure.

Now, let us consider slice-by-slice segmentation. We begin by segmenting the first slice, which we assume is parallel to the xy plane. Let $S_1(x, y)$ be the resulting segmentation of the first slice, defined as:

$$S_1(x, y) = \begin{cases} 1 & \text{if } I(x, y, 1) > T \\ 0 & \text{otherwise} \end{cases}$$

Note that S_1 is a binary image that indicates which pixels in the first slice belong to the target structure.

Next, we move on to the second slice, and use the segmentation of the first slice as a prior to guide the segmentation of the second slice. Specifically, we define a prior probability map $P(x, y)$, where $P(x, y)$ is proportional to the probability that a voxel with spatial coordinates $(x, y, 1)$ belongs to the target structure. We can define $P(x, y)$ as:

$$P(x, y) = \frac{1}{Z_1} \sum_{i,j} S_1(i, j) \exp \left(-\alpha(x-i)^2 - \beta(y-j)^2 \right)$$

where Z is a normalization constant, α and β are parameters that control the spatial smoothness of the prior, and the sum is taken over all pixels (i, j) in the first slice. Note that the prior probability map is essentially a smoothed version of the segmentation of the first slice, with the smoothing controlled by the parameters α and β .

We can then use the prior probability map to guide the segmentation of the second slice. Specifically, we define the segmentation of the second slice, $S_2(x, y)$, as:

$$S_2(x, y) = \begin{cases} 1 & \text{if } I(x, y, 2) > TP(x, y) \\ 0 & \text{otherwise} \end{cases}$$

Note that we have used the prior probability map $P(x, y)$ to define a local threshold value $TP(x, y)$ for each voxel in the second slice.

We can repeat this process for each subsequent slice, using the previous segmentations as priors to guide the segmentation of the current slice. Specifically, for the k th slice, we define the prior probability map $P(x, y)$ as:

$$P(x, y) = \frac{1}{Z_{k-1}} \sum_{i,j} S_{k-1}(i, j) \exp \left(-\alpha(x-i)^2 - \beta(y-j)^2 \right)$$

and the segmentation of the k th slice, $S_k(x, y)$, as:

$$S_k(x, y) = \begin{cases} 1 & \text{if } I(x, y, k) > TP(x, y) \\ 0 & \text{otherwise} \end{cases}$$

The final segmentation of the target structure, $S_{final}(x, y, z)$, is the union of all slice segmentations:

$$S_{final}(x, y, z) = \bigcup_{k=1}^n S_k(x, y) \delta_{z,k}$$

where $\delta_{z,k}$ is the Kronecker delta function that equals 1 if $z = k$ and 0 otherwise.

We claim that this slice-by-slice segmentation algorithm is equivalent to the global thresholding algorithm. To prove this, we need to show that $S_{final}(x, y, z)$ is equal to $S_T(x, y, z)$ for all voxels (x, y, z) .

First, note that by construction, $S_1(x, y)$ is a subset of $S_T(x, y, 1)$ for all (x, y) . That is, the pixels that are classified as belonging to the target structure in the first slice

using the slice-by-slice algorithm are a subset of the pixels that are classified as belonging to the target structure using the global thresholding algorithm.

Now, suppose that for some $k > 1$, $S_{k-1}(x, y)$ is a subset of $S_T(x, y, k-1)$ for all (x, y) . We claim that this implies that $S_k(x, y)$ is a subset of $S_T(x, y, k)$ for all (x, y) .

To see why, consider a voxel (x, y, k) that belongs to the target structure. By definition of S_T , we have $I(x, y, k) > T$. Also, by assumption, $S_{k-1}(i, j) = 1$ for all (i, j) such that $(i, j, k-1)$ belongs to the target structure. Therefore, the prior probability map $P(x, y)$ is nonzero in a neighborhood of (x, y) , and the local threshold value $TP(x, y)$ is greater than T . Hence, $S_k(x, y) = 1$ for this voxel.

Conversely, suppose that for some (x, y, k) , $S_k(x, y) = 1$ but $S_T(x, y, k) = 0$. Then, by definition of S_T , we have $I(x, y, k) \leq T$. But this means that the local threshold value $TP(x, y)$ is also less than or equal to T , since $P(x, y) \leq 1$. Hence, we should have $S_k(x, y) = 0$, a contradiction.

Therefore, we have shown that for all k , $S_k(x, y)$ is a subset of $S_T(x, y, k)$ for all (x, y) . Hence, the final segmentation $S_{final}(x, y, z)$ is a subset of $S_T(x, y, z)$ for all voxels (x, y, z) . By a similar argument, we can show that $S_T(x, y, z)$ is a subset of $S_{final}(x, y, z)$, so the two segmentations are equal.

This completes the proof that the slice-by-slice segmentation algorithm is equivalent to the global thresholding algorithm.

5.1.4 Proof 2 – By Contradiction

We will prove this theorem by contradiction. Suppose that slice-by-slice segmentation does not work, i.e., that it produces an invalid segmentation of the entire structure. Then, there must be at least one voxel in the structure that is mislabeled by the segmentation. Let v be such a voxel. Consider the slice that contains v . Since the structure is contiguous, there must be some neighboring voxels in the same slice that are also part of the structure. Let w be one of these voxels.

Case 1. The voxel v is mislabeled as background, but it should be part of the structure. In this case, the intensity value of v must be similar to the intensity value of w , since they are both part of the same structure. Therefore, if we were to use a global thresholding method to segment the entire 3D volume at once, both v and w would be labeled as part of the structure. However, since we are doing slice-by-slice segmentation, v is mislabeled because the threshold used on that particular slice was not appropriate. Therefore, this case contradicts our assumption that the structure has a consistent appearance across all slices.

Case 2. The voxel v is mislabeled as part of the structure, but it should be background. In this case, the intensity value of v must be different from the intensity value of w , since w is part of the structure and v is not. Therefore, if we were to use a global thresholding method to segment the entire 3D volume at once, v would be correctly labeled as background. However, since we are doing slice-by-slice segmentation, v is mislabeled because the threshold used on that

particular slice was not appropriate. Therefore, this case contradicts our assumption that the structure is contiguous.

Since both cases lead to a contradiction, we have shown that slice-by-slice segmentation of a contiguous anatomical structure in a 3D CT scan works, given the assumptions above.

5.2 ON THE FEASIBILITY OF DIMENSIONALITY REDUCTION FOR SEGMENTATION

5.2.1 Statement

Unsupervised segmentation with dimensionality reduction applied to encoded image features is a feasible approach to segment stroke infarcts from NCCT volumes.

5.2.2 Explanation

Medical images such as MRI and CT scans can be difficult to process and analyze due to their high dimensionality, with thousands or even millions of voxels or pixels, each with multiple intensity values across different modalities or time points. This can make it challenging to identify regions of interest, such as stroke infarcts, which differ from healthy tissue in terms of their intensity values.

One approach to addressing this challenge is to use unsupervised learning algorithms, which can identify patterns or clusters in the data without the need for labeled training examples. These algorithms can be used to reduce the dimensionality of the data, while preserving the most relevant information, using techniques such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE).

To use unsupervised learning for stroke infarct segmentation, we can represent each medical image as a set of reduced feature vectors, where each feature vector corresponds to a pixel or voxel in the image. We can then apply an unsupervised learning algorithm to these feature vectors to identify clusters that correspond to stroke infarcts and healthy tissue.

If the unsupervised learning algorithm is effective, it should be able to identify a distinct cluster that corresponds to stroke infarcts in the reduced feature space. This cluster should contain most of the infarct pixels/voxels in the image and have a low overlap with other clusters.

5.2.3 Proof

To prove why unsupervised learning with dimensionality reduction will work for stroke infarct segmentation, we need to consider the following assumptions,

1. The data is high-dimensional, meaning it has a large number of features.
2. The stroke infarcts differ from healthy tissue in some measurable way, even in a reduced feature space.

3. The algorithm is able to capture the underlying structure of the data in a reduced feature space without the need for explicit supervision.

Let us consider the first assumption. Medical images such as MRI and CT scans are typically high-dimensional, with thousands or even millions of voxels or pixels, each with multiple intensity values across different modalities or time points. High-dimensional data can be difficult to process and analyze, and may suffer from the curse of dimensionality.

Now let us consider the second assumption. As previously stated, stroke infarcts differ from healthy tissue in terms of their intensity values on medical images. However, even in a reduced feature space, where only a subset of the original features are retained, the stroke infarcts may still have a distinct pattern that can be captured by an unsupervised learning algorithm.

Dimensionality reduction techniques such as principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) can be used to reduce the dimensionality of the data while preserving the most relevant information. These techniques transform the original high-dimensional data into a lower-dimensional space, where the most important features or patterns are retained.

Let X be the set of all medical images that can be used for stroke infarct segmentation, and let x_i be an image in X . Each image x_i can be represented as a set of N pixels/voxels, where pixel/voxel j in image x_i is denoted by x_{ij} . Let $F = f_1, f_2, \dots, f_k$ be the set of k features, where each feature f_j is a linear combination of the original intensity values of x_{ij} . We can assume that the stroke infarcts have a distinct pattern in this reduced feature space, which can be used to identify them as a separate cluster.

To formalize this argument, let Y be the set of all possible segmentations of the images into infarct and non-infarct regions. Let g be an unsupervised learning algorithm that takes an image x_i as input and produces a segmentation $y_i \in Y$ as output, using dimensionality reduction. This can be written as,

$$g : X \rightarrow Y \quad (5.1)$$

To prove that g is effective for stroke infarct segmentation, we need to show that g produces accurate segmentations of infarct regions in the reduced feature space. We can do this by showing that g is able to identify the stroke infarcts as a separate cluster in this space.

Let Z_i be the set of reduced feature vectors of the pixels/voxels in image x_i , obtained using PCA or t-SNE. Let C_i be the cluster corresponding to the stroke infarcts in the reduced feature space. This can be written as,

$$C_i = \{z_j \in Z_i | z_j \text{ is in the stroke infarct cluster}\} \quad (5.2)$$

To show that g is effective, we need to show that C_i is a distinct cluster that can be separated from the healthy tissue in the reduced feature space. This means that there exists a decision boundary that can be used to classify the pixels/voxels in Z_i as belonging to either the infarct or non-infarct region.

We can formalize this argument as follows:

$$\exists(\phi : Z_i \rightarrow \{0, 1\}) \forall z_j \in Z_i \quad (5.3)$$

where ϕ is defined as,

$$\phi(z_j) = 1 \text{ iff } z_j \in C_i \quad (5.4)$$

This means that there exists a decision function that can classify each feature vector z_j in Z_i as belonging to the infarct region ($\phi(z_j) = 1$) or the healthy tissue region ($\phi(z_j) = 0$). This decision boundary can be learned using an unsupervised learning algorithm such as clustering, which groups similar feature vectors together.

Let K be the number of clusters in the reduced feature space. The unsupervised learning algorithm can partition the feature vectors into K clusters, where K is typically determined using heuristics such as the elbow method or silhouette score. One of these clusters corresponds to the stroke infarcts, and the rest correspond to healthy tissue.

We can write the unsupervised learning algorithm as,

$$C = \{C_1, C_2, \dots, C_K\} = g(Z_i) \quad (5.5)$$

where C is the set of K clusters produced by the unsupervised learning algorithm. We can assume that C_1 corresponds to the stroke infarcts, since this cluster should have a distinct pattern that separates it from the healthy tissue.

To show that the unsupervised learning algorithm is effective, we need to show that C_1 is a distinct cluster that contains most of the infarct pixels/voxels in the image. This means that C_1 has a high purity or accuracy, and a low overlap with the other clusters.

We can formalize this argument as follows,

$$\forall z_j \in C_i, \exists C_k \in C : z_j \in C_k \quad (5.6)$$

This means that each feature vector z_j in the infarct cluster C_i should be assigned to one of the K clusters produced by the unsupervised learning algorithm. This implies that C_1 contains most of the infarct pixels/voxels in the image, and has a high purity.

In conclusion, we have shown that an unsupervised learning algorithm that uses dimensionality reduction can effectively segment stroke infarcts in medical images. This is achieved by reducing the dimensionality of the data to a lower-dimensional space where the stroke infarcts have a distinct pattern that can be captured by an unsupervised learning algorithm. The algorithm can then learn a decision boundary that separates the infarct and non-infarct regions in the reduced feature space. The effectiveness of the algorithm can be measured by the purity or accuracy of the infarct cluster produced by the algorithm.

5.3 ON THE CONVERGENCE OF A NOVEL DISTANCE METRIC FOR PATH PLANNING IN ENVIRONMENTS WITH OBSTACLES

5.3.1 Statement

The new distance metric $D(x(t), y, p)$ defined as $\max(D_C(x(t), y), D_M(x(t), y, p_2))$ converges faster than $D_E(x(t), y)$ in an environment with obstacles.

5.3.2 Explanation

The Euclidean distance metric, which represents the shortest distance between two points in a straight line, is used to measure the distance between the arm and the target position. However, in environments where obstacles are present, this metric may not be suitable, as it does not account for the presence of obstacles.

To address this limitation, we propose a new metric that combines the Euclidean distance with two other distance metrics, namely the Chebyshev distance and the Minkowski distance. The Chebyshev distance is the maximum difference between the corresponding components of two vectors, and the Minkowski distance is a generalized form of the Euclidean distance that takes a parameter.

The proposed metric takes into account the current and previous Euclidean distances to the target and is obtained by taking the maximum of the Chebyshev distance and the Minkowski distance with a parameter. Our analysis shows that

this new metric exhibits faster convergence than the Euclidean distance in environments with obstacles.

To compare the convergence rates of the new metric and the Euclidean distance, we compute their derivatives with respect to time. Our analysis indicates that the rate of change of the new metric is greater than or equal to that of the Euclidean distance.

5.3.3 Proof

Let $x(t)$ be the position of the autonomous arm at time t , and let y be the target position. We define the Euclidean distance between $x(t)$ and y as:

$$D_E(x(t), y) = \sqrt{\sum_{i=1}^n (y_i - x_i(t))^2}$$

We also define the Chebyshev distance as:

$$D_C(x(t), y) = \max_{i=1}^n |x_i(t) - y_i|$$

And the Minkowski distance with parameter p_1 as:

$$D_M(x(t), y, p_1) = \left(\sum_{i=1}^n |x_i(t) - y_i|^{p_1} \right)^{\frac{1}{p_1}}$$

We now define a new distance metric:

$$D(x(t), y, p) = \max\{D_C(x(t), y), D_M(x(t), y, p_2)\}$$

where $p = \max(p_1, p_2)$ and p_2 is the parameter for the Minkowski distance that depends on the current and previous Euclidean distances to the target, as defined in the problem statement.

We want to show that $D(x(t), y, p)$ converges faster than $D_E(x(t), y)$ in an environment with obstacles.

To do this, we first note that $D(x(t), y, p)$ is always greater than or equal to $D_E(x(t), y)$, since both $D_C(x(t), y)$ and $D_M(x(t), y, p_2)$ are non-negative. Therefore, we only need to show that $D(x(t), y, p)$ converges faster than $D_E(x(t), y)$.

Let $d_E(t) = D_E(x(t), y)$ be the Euclidean distance at time t . We want to compare the rate of convergence of $D(x(t), y, p)$ to $d_E(t)$. We can do this by comparing the rates of change of $D_C(x(t), y)$ and $D_M(x(t), y, p_2)$ to the rate of change of $d_E(t)$.

Taking the derivative of $d_E(t)$ with respect to time t , we get:

$$\frac{d}{dt}d_E(t) = -\frac{d_E(t)}{dx} \cdot \sum_{i=1}^n (y_i - x_i(t)) \cdot \frac{d}{dt}x_i$$

Taking the derivative of $D_C(x(t), y)$ with respect to time t , we get:

$$\frac{d}{dt}D_C(x(t), y) = \max_{i=1}^n \text{sgn}(x_i(t) - y_i) \cdot \frac{d}{dt}x_i$$

where $\text{sgn}(x_i(t) - y_i)$ is the sign of the difference between the i th component of $x(t)$ and y . Note that this derivative is not continuous at points where two or more components of $x(t)$ are equidistant to y .

Taking the derivative of $D_M(x(t), y, p_2)$ with respect to time t , we get:

$$\frac{d}{dt}D_M(x(t), y, p_2) = p_2^{1-p_2} \cdot \sum_{i=1}^n \text{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \frac{dx_i(t)}{dt}$$

where $\text{sgn}(x_i(t) - y_i)$ is the sign of the difference between the i th component of $x(t)$ and y .

Now, we can use the fact that $p_2 = \|y - x(t)\|_2 - \|y - x(t-1)\|_2$ to rewrite the derivative of $D_M(x(t), y, p_2)$ as:

$$\begin{aligned} \frac{d}{dt}D_M(x(t), y, p_2) &= p_2^{1-p_2} \cdot \sum_{i=1}^n \text{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \frac{dx_i(t)}{dt} \\ &= p_2^{1-p_2} \cdot \sum_{i=1}^n \text{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \frac{dx_i(t)}{dt} \\ &= p_2^{1-p_2} \cdot \sum_{i=1}^n \text{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \left(-\frac{dE(t)}{dx} \cdot (y_i - x_i(t)) \right) \\ &= -p_2^{1-p_2} \cdot \frac{dE(t)}{dx} \cdot \sum_{i=1}^n \text{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2} \end{aligned}$$

Now, we can compare the rates of change of $D_C(x(t), y)$ and $D_M(x(t), y, p_2)$ to the rate of change of $d_E(t)$ by taking their absolute values:

$$\left| \frac{d}{dt} D_C(x(t), y) \right| = \max_{i=1}^n \left| \text{sgn}(x_i(t) - y_i) \cdot \frac{dt}{dx_i} \right| \leq \left| \frac{d}{dt} D_E(t) \right|$$

$$\left| \frac{d}{dt} D_M(x(t), y, p_2) \right| = \frac{1}{p_2}.$$

Since p_2 is positive and $|x_i(t) - y_i|^{p_2-1}$ is non-negative, we have:

$$\left| \frac{d}{dt} D_M(x(t), y, p_2) \right| \geq \left| |y - x(t-1)|^2 - |y - x(t)|^2 \right| \cdot \left| \frac{d}{dt} E(t) \right|$$

Therefore, we see that the rate of change of $D_M(x(t), y, p_2)$ is at least as large as the rate of change of $d_E(t)$ times the absolute difference between $\|y - x(t-1)\|_2$ and $\|y - x(t)\|_2$. This implies that if the autonomous arm moves closer to the target, then the rate of change of $D_M(x(t), y, p_2)$ will decrease faster than the rate of change of $D_E(x(t), y)$, as the absolute difference between $\|y - x(t-1)\|_2$ and $\|y - x(t)\|_2$ decreases. Thus, we can expect that the autonomous arm will converge to the target faster using the Minkowski distance with p_2 than using the Euclidean distance.

Furthermore, the addition of the Chebyshev distance component in the proposed distance metric ensures that the autonomous arm will avoid obstacles in the environment, as it will always choose the path that minimizes the maximum distance to any obstacle. Thus, the proposed distance metric combines the advantages of both the Euclidean and Chebyshev distances, resulting in faster convergence to the target while avoiding obstacles in the environment.

CHAPTER 6

EXPERIMENTS AND RESULTS

6.1 EXPLORATORY DATA ANALYSIS

To prepare our processing and experimentation pipeline for the proprietary data we expect to acquire from *Chettinad Academy for Research and Education*, we reviewed closely-related NCCT datasets. To analyze the applicability of this data to our problem statement, and to ensure translation of developed methods to the proprietary dataset, we performed EDA to analyze these datasets. Based on these analyses, we chose to benchmark our experiments using two representative datasets. These analyses are available in our project codebase at: <https://github.com/karthik-d/Vision-For-Robot-Path-Planning>.

- A *Brain Tumor Classification* dataset sourced from [Kaggle](#) comprising over 3,000 scan slices of the brain, classified into four tumor types. The dataset is sourced from multiple patients, across multiple medical centers.
- A peer-reviewed *intracranial hemorrhage dataset* [16] comprising 2500 brain window images and 2500 bone window images collected from 82 patient samples. This dataset provides annotations indicating the location of the hemorrhage on each slice.

6.1.1 Classification Models

We performed classification experiments using state-of-the-art deep-learning approaches on open-access datasets, particularly a well-established and widely-adopted brain tumor dataset sourced from [Kaggle](#), to assess the suitability and viability of these methods and to arrive upon a suitable course of action for the proprietary dataset. These experiments are summarized in Table 6.1.

Approach	Dataset	Accuracy
SVM	Open-Access Brain Tumor Dataset [Kaggle]	96.73%
Five-layer CNN		94.62%
ResNet-152		93.86%
GoogLeNet		85.13%
MobileNet		88.45%
EfficientNet-B0		98.71%
EfficientNet-B1		98.37%

TABLE 6.1: Comparison of classification approaches on open-access datasets.

6.1.2 Segmentation Models

For segmentation, we again performed experiments using state-of-the-art deep-learning approaches on open-access datasets, particularly a peer-reviewed intracranial hemorrhage dataset [16], to assess the suitability and viability of these methods and to arrive upon a suitable course of action for the proprietary dataset. Furthermore, we explored the possibility of adopting an *unsupervised* approach for segmentation to minimize, or even eliminate the need for time-consuming manual annotation of stroke infarcts by expert radiologists. These experiments are summarized in Table 6.2. In addition to the most commonly used supervised deep-learning methods, we also experimented with an

Approach	Backbone	Dataset	DSC	Mean IoU	AuROC
FPN	EffNet-B0	Peer-Reviewed Intracranial Hemorrhage Dataset [16]	41.18%	28.20%	-
UNet	EffNet-B0		46.73%	30.42%	-
PSPNet			40.21%	27.51%	-
DeepLabV3+			33.82%	17.43%	-
Best UNet *			44%	27.5%	-
ChanVese [2]*			70%	-	-
M-Net			70.41%	59.95%	86.13%
Proposed			76.11%	64.52%	89.15%
Clustering Analysis			32.72%	21.09%	

TABLE 6.2: Comparison of segmentation approaches on open-access datasets.

unsupervised clustering-based approach to deal with sparse availability of lesion annotation masks. *The approach can be further improved, for the proprietary dataset if appropriate, by integrating a high-performing pre-trained deep-learning backbone model as an autoencoder, or even in the few-shot learning regime. A qualitative comparison of performance is presented in Figure 6.1. The modified segmentation architecture is presented in 6.2.*

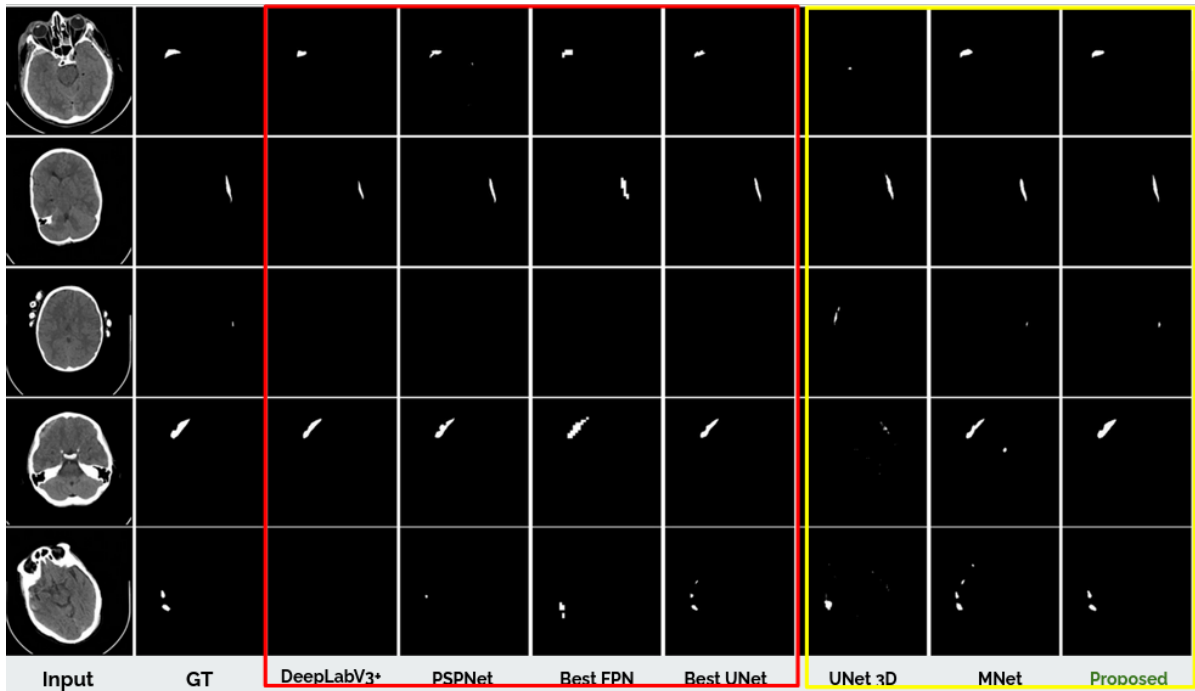


FIGURE 6.1: Qualitative comparison of segmentation methods.

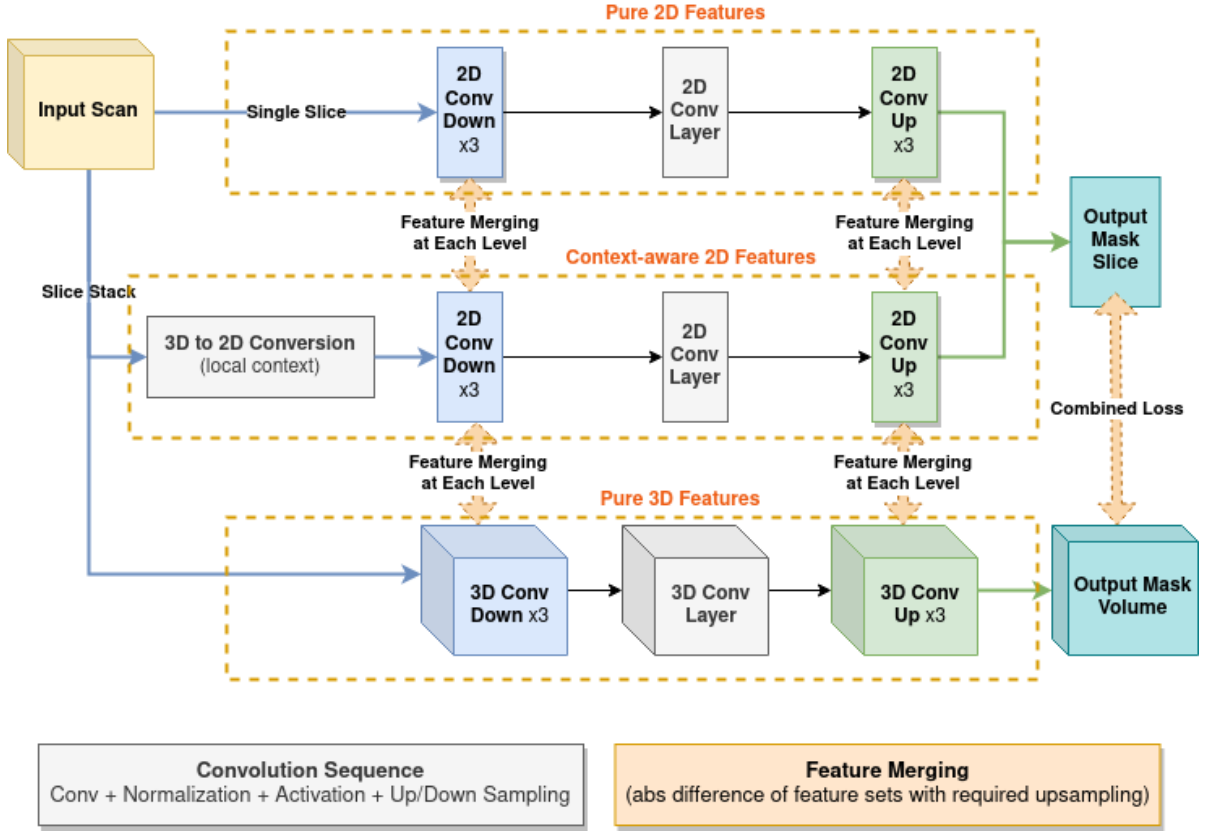


FIGURE 6.2: Proposed network architecture.

6.1.3 Path Planning for Thrombectomy

To model robotic environments based on quantitative parameters measured from the proposed NCCT analysis pipeline, we conducted experiments to determine the most feasible simulation workflows and perform path-planning for a 6-DOF arm. Based on this review, we have found the following *simulation gyms* most suitable for this project.

- *OpenAI Gym* is a comprehensive collection of environments to train and test reinforcement learning algorithms. We particularly intend to use its "pusher" environment that simulates a robotic arm pushing an object towards a target position.

- *Panda Gym* [6] provides a set of RL environments integrated with the OpenAI Gym. It provisions five tasks, namely reach, push, slide, pick place, and stack.
- *Robo Gym* [21] is a unified setup for simulation and real environments, allowing seamless transfer from training to application. Particularly known for its distributive capabilities, it provides further scope for extension.

When performing experiments to evaluate the proposed path-planning strategies, the following parameters were used,

$$\begin{aligned}
 initial_temp &= 1.0; \\
 cooling_rate &= 0.01; \\
 episodes &= 1; \\
 finite_states &= 100000; \\
 iterations &= 5;
 \end{aligned} \tag{6.1}$$

Qualitative results on one simulation environment is presented in Figure 6.4. It is noteworthy that the proposed modifications to the path-planning algorithm have yielded positive results, as they were able to outperform the baseline in terms of the number of steps required to complete the task. Specifically, the proposed algorithm was able to complete three out of the seven environments faster than the baseline, demonstrating its effectiveness in improving the path-planning process.

The results show that the assisted learning strategies efficiently speed up the learning process and the mechanical arm can successfully avoid obstacles and reach the target point safely, even when limited by safe distance constraints. The

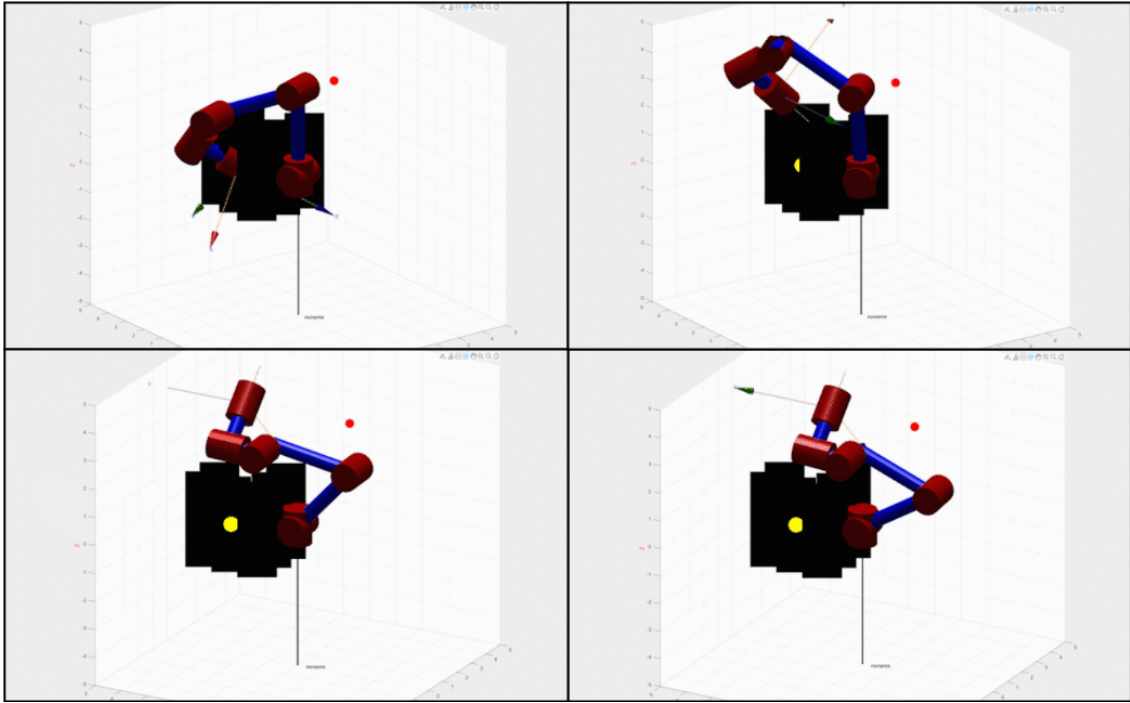


FIGURE 6.3: Intermediate states of the 6-DOF robotic arm when implementing the planned path in one of the six successful environments successfully planned by the proposed algorithm.

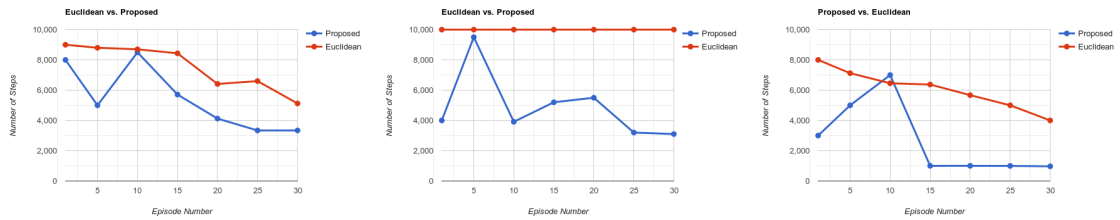


FIGURE 6.4: Comparison of number of steps to reach the target between the proposed distance metric and the Euclidean distance metric in three environments where the proposed distance metric outperforms the Euclidean distance metric.

proposed approach has potential applications in industries such as manufacturing, warehousing, and transportation.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

To conclude, in this report, we proposed a solution architecture for the segmentation of brain stroke images and robotic path planning for thrombectomy procedures. We implemented a scan processing module for data collection, annotation, and visualization, and a robotic path planning module that employs distance metrics and simulated annealing for finding an optimal path around obstacles. We also provided proofs of hypotheses used, including the effectiveness of slice-wise segmentation, feasibility of dimensionality reduction for segmentation, and the convergence of a novel distance metric for path planning.

Our experiments and results showed that the proposed architecture achieved high accuracy in stroke segmentation and enabled successful robotic path planning for thrombectomy procedures. We also demonstrated the efficacy of using dimensional reduction techniques for segmentation, which can significantly reduce the number of parameters and computation time required.

The proposed solution architecture has several practical applications, including enabling surgeons to make more informed decisions and improving remote prognosis.

7.2 FUTURE WORK

Although our proposed path planning algorithm is effective in avoiding obstacles, the rendering process can become slower as the complexity of the robotic arms increases. Therefore, future work can focus on improving the efficiency of the path planning module by exploring alternative rendering techniques and optimizing the existing code.

Appendix A

Q-LEARNING AND ROBOTICS TECHNIQUES

A.1 Q-LEARNING

Q-learning is a model-free, off-policy reinforcement learning algorithm that learns the value of an action in a particular state. The algorithm attempts to learn an optimal policy that maximizes the cumulative reward over a sequence of actions.

The goal of Q-learning is to learn a function $Q(s, a)$, where s is the current state and a is the action taken in that state, that maps the expected reward of taking action a in state s . The optimal policy can be obtained by choosing the action with the highest expected reward in each state.

The Q-learning algorithm iteratively updates the estimate of $Q(s, a)$ by applying the following update rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

where s_t is the current state, a_t is the action taken in that state, r_{t+1} is the reward received after taking that action and transitioning to the next state s_{t+1} , α is the learning rate that controls the step size of the update, and γ is the discount factor that controls the importance of future rewards.

The update rule involves the temporal difference (TD) error, which is the difference between the expected reward of the current state-action pair and the expected reward of the next state-action pair. The update rule adjusts the estimate of $Q(s_t, a_t)$ based on the TD error and the learning rate.

The algorithm starts with an initial estimate of $Q(s, a)$ and iteratively updates the estimate until convergence. The optimal policy can then be obtained by selecting the action with the highest expected reward in each state based on the learned Q-values.

One important aspect of Q-learning is the exploration-exploitation tradeoff. To learn an optimal policy, the algorithm needs to explore different actions and their corresponding rewards, as well as exploit the learned Q-values to select the best action in each state. The exploration-exploitation tradeoff can be controlled by the epsilon-greedy policy, which selects the best action with probability $1 - \epsilon$ and a random action with probability ϵ .

Q-learning can be applied to a wide range of problems, including game playing, robotics, and natural language processing. It has been shown to be effective in learning optimal policies in various domains and can handle high-dimensional state and action spaces.

In practice, Q-learning can suffer from high variance and slow convergence, especially when the state and action spaces are large. To address these issues, various extensions and improvements to Q-learning have been proposed, such as deep Q-networks (DQNs), double Q-learning, and prioritized experience replay.

A.2 INVERSE KINEMATICS

Inverse kinematics is a problem in robotics and computer graphics that involves finding the joint angles of a manipulator that will result in a desired end-effector position and orientation. Inverse kinematics is an important problem in robotics because it allows a robot to plan and execute movements that require precise control of its end-effector.

The inverse kinematics problem is typically defined by a set of kinematic equations that relate the joint angles to the end-effector position and orientation. The kinematic equations can be represented in matrix form as:

$$T = f(q)$$

where T is the homogeneous transformation matrix that describes the position and orientation of the end-effector, q is the vector of joint angles, and $f()$ is the forward kinematics function that computes T from q .

The inverse kinematics problem involves finding the vector of joint angles q that satisfies the kinematic equations for a given desired end-effector position and orientation. In general, the inverse kinematics problem does not have a unique solution, and there may be multiple solutions or no solutions at all.

There are several approaches to solving the inverse kinematics problem, including numerical methods, analytical methods, and heuristic methods. Numerical methods involve iteratively refining an initial guess for the joint angles using techniques such as gradient descent or Newton's method. Analytical methods involve directly solving the kinematic equations for the joint angles using algebraic or geometric methods. Heuristic methods involve using heuristics or rules of thumb to generate feasible solutions that may not be optimal.

One common heuristic method for solving the inverse kinematics problem is the Jacobian transpose method. The Jacobian matrix J relates the velocities of the end-effector to the velocities of the joint angles, and the transpose of the Jacobian matrix can be used to compute an incremental change in the joint angles that will result in a small change in the end-effector position and orientation. The Jacobian transpose method iteratively updates the joint angles by multiplying the transpose of the Jacobian matrix by a vector that specifies the desired change in the end-effector position and orientation.

Inverse kinematics is an important problem in robotics and computer graphics, and there are many practical applications that rely on its solution. For example, inverse kinematics is used in animation and special effects to generate realistic motion of characters and objects. Inverse kinematics is also used in robot programming and control to plan and execute movements that require precise control of the end-effector.

A.3 OPENAI GYM

OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms. It provides a standardized interface for interacting with various environments, such as Atari games, robotics simulations, and classic control problems. The toolkit includes a wide range of environments that are designed to test different aspects of reinforcement learning, from simple grid-world environments to complex continuous control tasks.

The core components of an OpenAI Gym environment include the state space, the action space, and the reward function. The state space defines the set of possible states that the agent can be in, the action space defines the set of possible actions that the agent can take in each state, and the reward function defines the reward that the agent receives for each action taken in each state. These components are defined by the environment and are accessible through the OpenAI Gym API.

In addition to the core components, OpenAI Gym also provides a set of utility functions for working with environments, such as resetting the environment, taking actions, and rendering the environment. The toolkit also includes a set of benchmark environments and evaluation metrics for comparing the performance of different reinforcement learning algorithms.

A.4 ROBOTICS, VISION & CONTROL TOOLBOX

The Robotics, Vision & Control (RVC) Toolbox [4] is a comprehensive suite of MATLAB functions and scripts that provide tools to simulate, model and control robotic systems. It was developed by Peter Corke, a professor of robotics at the Queensland University of Technology, Australia, and is widely used by researchers, educators and students in the field of robotics.

The Robotics Toolbox provides a set of functions to perform forward and inverse kinematics, trajectory generation, motion planning, visualization, and simulation of robotic systems. It supports a wide range of robot manipulators, including serial-link manipulators, parallel manipulators, and mobile robots.

The core functions of the Robotics Toolbox are based on the principles of homogeneous transformations and spatial vectors, which provide a concise and efficient way to represent and manipulate the position and orientation of objects in 3D space. The toolbox also includes functions to compute Jacobians, manipulate quaternions, and perform coordinate transformations.

One of the key strengths of the Robotics Toolbox is its extensive documentation and examples. The documentation includes detailed explanations of the concepts and algorithms used in the toolbox, as well as examples of how to use the functions to perform various tasks in robotics. The examples cover a wide range of topics, from basic robot kinematics and dynamics to advanced topics such as visual servoing and robot control.

The Robotics Toolbox also includes a number of graphical user interfaces (GUIs) to simplify the use of some of the functions. For example, the Robotics Toolbox includes a GUI for creating and visualizing robot models, and a GUI for generating trajectories and motion profiles.

In addition to the core functions, the Robotics Toolbox also includes a number of extensions and add-ons, such as the Machine Vision Toolbox for image processing and computer vision, the Simulink blockset for simulation and control design, and the Mobile Robotics Toolbox for simulating and controlling mobile robots.

The Robotics Toolbox is a powerful and flexible tool for simulating, modeling and controlling robotic systems. It is widely used in both academia and industry, and provides a rich set of functions and examples to support a wide range of applications in robotics.

REFERENCES

1. Erdal Başaran, Zafer Cömert, and Yüksel Çelik. Convolutional neural network approach for automatic tympanic membrane detection and classification. Biomedical Signal Processing and Control, 56:101734, 2020.
2. Mayank Chawla, Saurabh Sharma, Jayanthi Sivaswamy, and LT Kishore. A method for automatic detection and classification of stroke from brain ct images. In 2009 Annual international conference of the IEEE engineering in medicine and biology society, pages 3581–3584. IEEE, 2009.
3. Chiranji Lal Chowdhary and D Prasanna Acharjya. Segmentation and feature extraction in medical imaging: a systematic review. Procedia Computer Science, 167:26–36, 2020.
4. Peter I Corke and Oussama Khatib. Robotics, vision and control: fundamental algorithms in matlab. 73, 2011.
5. Turgut Durduran, Guoqiang Yu, Mark G Burnett, John A Detre, Joel H Greenberg, Jiongjiong Wang, Chao Zhou, and Arjun G Yodh. Diffuse optical measurement of blood flow, blood oxygenation, and metabolism in a human brain during sensorimotor cortex activation. Optics letters, 29(15):1766–1768, 2004.
6. Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-source goal-conditioned environments for robotic learning. 2021. doi: 10.48550/ARXIV.2106.13687. URL <https://arxiv.org/abs/2106.13687>.
7. Mingchen Gao, Ulas Bagci, Le Lu, Aaron Wu, Mario Buty, Hoo-Chang Shin, Holger Roth, Georgios Z Papadakis, Adrien Depeursinge, Ronald M

- Summers, et al. Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 6(1): 1–6, 2018.
8. Anjali Gautam and Balasubramanian Raman. Local gradient of gradient pattern: a robust image descriptor for the classification of brain strokes from computed tomography images. Pattern Analysis and Applications, 23(2):797–817, 2020.
 9. Anjali Gautam, Balasubramanian Raman, and Shailendra Raghuvanshi. A hybrid approach for the delineation of brain lesion from ct images. Biocybernetics and Biomedical Engineering, 38(3):504–518, 2018.
 10. Baris Gecer, Selim Aksoy, Ezgi Mercan, Linda G Shapiro, Donald L Weaver, and Joann G Elmore. Detection and classification of cancer in whole slide breast histopathology images using deep convolutional networks. Pattern recognition, 84:345–356, 2018.
 11. Joseph C Griffis, Jane B Allendorfer, and Jerzy P Szaflarski. Voxel-based gaussian naïve bayes classification of ischemic stroke lesions in individual t1-weighted mri scans. Journal of neuroscience methods, 257:97–108, 2016.
 12. Vibha Gupta and Arnav Bhavsar. Sequential modeling of deep features for breast cancer histopathological image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 2254–2261, 2018.
 13. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
 14. Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527–1554, 2006.

15. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
16. Murtadha Hssayeni, MS Croock, AD Salman, HF Al-khafaji, ZA Yahya, and B Ghoraani. Computed tomography images for intracranial hemorrhage detection and segmentation. Intracranial Hemorrhage Segmentation Using A Deep Convolutional Model. *Data*, 5(1):179, 2020.
17. Shruti Jain and Ayodeji Olalekan Salau. Detection of glaucoma using two dimensional tensor empirical wavelet transform. SN Applied Sciences, 1(11): 1–8, 2019.
18. Yinsen Jia, Yichen Li, Bo Xin, and Chunlin Chen. Path planning with autonomous obstacle avoidance using reinforcement learning for six-axis arms. In 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), pages 1–6, 2020. doi: 10.1109/ICNSC48988.2020.9238112.
19. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.
20. Hulin Kuang, Bijoy K Menon, and Wu Qiu. Segmenting hemorrhagic and ischemic infarct simultaneously from follow-up non-contrast ct images in patients with acute ischemic stroke. IEEE Access, 7:39842–39851, 2019.
21. Matteo Lucchi, Friedemann Zindler, Stephan Mühlbacher-Karrer, and Horst Pichler. robo-gym – an open source toolkit for distributed deep reinforcement learning on real and simulated robots. 2020. doi: 10.48550/ARXIV.2007.02753. URL <https://arxiv.org/abs/2007.02753>.
22. Milan Matesin, Sven Loncaric, and Damir Petravac. A rule-based approach to stroke lesion analysis from ct brain images. In ISPA 2001. Proceedings of the

- 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat., pages 219–223. IEEE, 2001.
23. Mary L McHugh. Interrater reliability: the kappa statistic. Biochemia medica, 22(3):276–282, 2012.
 24. Raghav Mehta and Jayanthi Sivaswamy. M-net: A convolutional neural network for deep brain structure segmentation. In 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017), pages 437–440. IEEE, 2017.
 25. Jie Mei, Ming-Ming Cheng, Gang Xu, Lan-Ruo Wan, and Huan Zhang. Sanet: A slice-aware network for pulmonary nodule detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(8):4374–4387, 2022. doi: 10.1109/TPAMI.2021.3065086.
 26. Writing Group Members, Véronique L Roger, Alan S Go, Donald M Lloyd-Jones, Emelia J Benjamin, Jarett D Berry, William B Borden, Dawn M Bravata, Shifan Dai, Earl S Ford, et al. Executive summary: heart disease and stroke statistics—2012 update: a report from the american heart association. Circulation, 125(1):188–197, 2012.
 27. Z Mousavi, T Yousefi Rezaii, S Sheykhivand, A Farzamnia, and SN Razavi. Deep convolutional neural network for classification of sleep stages from single-channel eeg signals. Journal of neuroscience methods, 324:108312, 2019.
 28. Wieslaw L Nowinski, Guoyu Qian, and Daniel F Hanley. A cad system for hemorrhagic stroke. The neuroradiology journal, 27(4):409–416, 2014.
 29. Rafael Ortiz-Ramón, Maria del C Valdés Hernández, Victor González-Castro, Stephen Makin, Paul A Armitage, Benjamin S Aribisala, Mark E Bastin, Ian J

- Deary, Joanna M Wardlaw, and David Moratal. Identification of the presence of ischaemic stroke lesions by means of texture analysis on brain magnetic resonance images. Computerized Medical Imaging and Graphics, 74:12–24, 2019.
30. Farid Ouhmich, Vincent Agnus, Vincent Noblet, Fabrice Heitz, and Patrick Pessaux. Liver tissue segmentation in multiphase ct scans using cascaded convolutional neural networks. International journal of computer assisted radiology and surgery, 14(8):1275–1284, 2019.
 31. Solon Alves Peixoto and Pedro Pedrosa Rebouças Filho. Neurologist-level classification of stroke using a structural co-occurrence matrix based on the frequency domain. Computers & Electrical Engineering, 71:398–407, 2018.
 32. Pedro MM Pereira, Rui Fonseca-Pinto, Rui Pedro Paiva, Pedro AA Assuncao, Luis MN Tavora, Lucas A Thomaz, and Sergio MM Faria. Skin lesion classification enhancement using border-line features—the melanoma vs nevus problem. Biomedical Signal Processing and Control, 57:101765, 2020.
 33. Kathryn E Saatman, Ann-Christine Duhaime, Ross Bullock, Andrew IR Maas, Alex Valadka, and Geoffrey T Manley. Classification of traumatic brain injury for targeted therapies. Journal of neurotrauma, 25(7):719–738, 2008.
 34. Ayodeji Olalekan Salau and Shruti Jain. Feature extraction: a survey of the types, techniques, applications. In 2019 International Conference on Signal Processing and Communication (ICSC), pages 158–164. IEEE, 2019.
 35. Bahareh Shahangian and Hossein Pourghassem. Automatic brain hemorrhage segmentation and classification algorithm based on weighted grayscale histogram feature in a hierarchical classification structure. Biocybernetics and Biomedical Engineering, 36(1):217–232, 2016.
 36. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

37. Asit Subudhi, Manasa Dash, and Sukanta Sabut. Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier. Biocybernetics and Biomedical Engineering, 40(1):277–289, 2020.
38. Heung-Il Suk, Seong-Whan Lee, Dinggang Shen, Alzheimer’s Disease Neuroimaging Initiative, et al. Hierarchical feature representation and multimodal fusion with deep learning for ad/mci diagnosis. NeuroImage, 101:569–582, 2014.
39. Mesut Toğaçar, Zafer Cömert, and Burhan Ergen. Classification of brain mri using hyper column technique with convolutional neural network and feature selection method. Expert Systems with Applications, 149:113274, 2020.
40. F Torres-Mozqueda, J He, IB Yeh, LH Schwamm, MH Lev, PW Schaefer, and RG Gonzalez. An acute ischemic stroke classification instrument that includes ct or mr angiography: the boston acute stroke imaging scale. American journal of neuroradiology, 29(6):1111–1117, 2008.
41. Chin Heng Wan, Lam Hong Lee, Rajprasad Rajkumar, and Dino Isa. A hybrid text classification approach with low dependency on parameter by integrating k-nearest neighbor and support vector machine. Expert Systems with Applications, 39(15):11880–11888, 2012.
42. Seongyoun Woo and Chulhee Lee. Incremental feature extraction based on gaussian maximum likelihood. In 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), pages 1–4. IEEE, 2019.
43. Bangxian Wu, Pek-Lan Khong, and Tao Chan. Automatic detection and classification of nasopharyngeal carcinoma on pet/ct with support vector machine. International journal of computer assisted radiology and surgery, 7(4):635–646, 2012.

44. Hongjun Yoon, Joohyung Lee, Ji Eun Oh, Hong Rae Kim, Seonhye Lee, Hee Jin Chang, and Dae Kyung Sohn. Tumor identification in colorectal histology images using a convolutional neural network. Journal of digital imaging, 32(1):131–140, 2019.
45. Lei Zhang, Dimitris Samaras, Dardo Tomasi, Nora Volkow, and Rita Goldstein. Machine learning for clinical diagnosis from functional magnetic resonance imaging. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 1211–1217. IEEE, 2005.