2023

Integrating the Spatial Pyramid Pooling into 3D Convolutional Neural Networks for Cerebral Microbleeds Detection

Andre Accioly Veira

Integrating the Spatial Pyramid Pooling into 3D Convolutional Neural Networks for Cerebral Microbleeds Detection

by

Andre Accioly Vieira

A dissertation presented as partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in
Computer Science

College of Computing and Engineering
Nova Southeastern University

2023

We hereby certify that this dissertation, submitted by Andre Vieira conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

Sumitra Mukherjee

Sumitra Mukherjee, Ph.D.
Chairperson of Dissertation Committee

4/ 12 /23
Date

Michael Laszlo, Ph.D.
Dissertation Committee Member

4/ 12 /23
Date

Francisco J. Mitropoulos, Ph.D.
Dissertation Committee Member

4/ 12 /23
Date

Approved:

Meline Kevorkian

Meline Kevorkian, Ed.D.
Dean, College of Computing and Engineering

4/ 12 /23
Date

College of Computing and Engineering
Nova Southeastern University

2023

An Abstract of a Dissertation Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Integrating the Spatial Pyramid Pooling into 3D Convolutional Neural Networks for Cerebral Microbleeds Detection

By

Andre Accioly Vieira

March 2023

Cerebral microbleeds (CMB) are small foci of chronic blood products in brain tissues that are critical markers for cerebral amyloid angiopathy. CMB increases the risk of symptomatic intracerebral hemorrhage and ischemic stroke. CMB can also cause structural damage to brain tissues resulting in neurologic dysfunction, cognitive impairment, and dementia. Due to the paramagnetic properties of blood degradation products, CMB can be better visualized via susceptibility-weighted imaging (SWI) than magnetic resonance imaging (MRI).

CMB identification and classification have been based mainly on human visual identification of SWI features via shape, size, and intensity information. However, manual interpretation can be biased. Visual screening may miss small CMB or be confused by CMB mimics. Therefore, developing automatic methods for CMB detection is critical, and recent research has been directed at finding solutions based on automated feature extraction.

One of the most promising automated solutions uses a 3-dimensional convolutional neural network (3D-CNN) approach to screen and discriminate CMBs. The method uses an improved sliding window strategy, avoiding redundant computation and reducing the classification workload. However, despite its satisfactory results, the technique still suffers from limitations such as the lack of spatial information, poor handling of CMB size variation, and the existence of CMB mimics.

This dissertation implements the SPP strategy into the 3D CNN two-stage model. It investigates its advantages over extant methods for CMB detection to enhance its discrimination capabilities without compromising detection speeds. The suggested model improved the results obtained by the 3D-CNN method yielding an overall 96.94% sensitivity and 95.48% precision. Another contribution is providing the 3D-CNN-SPP with the capability to detect CMBs of different sizes and shapes.

Acknowledgments

As I write this acknowledgment for my doctorate in computer science, I feel my heart full of gratitude. This degree represents the culmination of many years of hard work, dedication, and support from many people, especially during the pandemic. I could not have achieved this accomplishment without the help and guidance of many individuals.

I want to take this opportunity to express my heartfelt appreciation to everyone who has contributed to my success.

First and foremost, I would like to thank my academic advisor, Dr. Sumitra Mukherjee, for providing me with the guidance, support, and inspiration needed to complete my dissertation. His insights, expertise, and patience were invaluable in shaping my research, and his unwavering support has been a source of strength and motivation for me.

I would also like to thank the members of my dissertation committee, Dr. Michael Laszlo, and Dr. Frank J. Mitropoulos, for their invaluable feedback and support during the dissertation defense process. Their constructive criticism, insightful comments, and suggestions helped refine and present my research in the best possible light.

I am deeply grateful to the College of Computing and Engineering faculty members at Nova Southeastern University, who have contributed to my academic growth. Their courses, seminars, and workshops have expanded my knowledge, sharpened my skills, and broadened my perspectives. I would also like to thank the staff members who have supported me countless times. Their assistance with administrative matters has been indispensable.

I want to express my heartfelt gratitude to my beloved wife, Patricia, whose support and encouragement have sustained me throughout my academic journey. Her love, patience, and understanding have been my anchor, even during the most challenging times. I am indebted for her sacrifices. Her belief in me has been a constant source of motivation and joy.

To my friends, I cannot thank you enough for your unwavering support, love, and encouragement throughout my academic journey.

Finally, I want to acknowledge the broader academic community, whose ideas and insights have informed and inspired my research. I am grateful to the many scholars whose seminal works have shaped my research interests and helped me appreciate the broader context of computer science.

In conclusion, I am deeply grateful to everyone who has played a role in my academic journey. Your support, guidance, and encouragement have been essential to my success, and I am honored to have earned this doctorate in computer science.

Table of Contents

Chapters

List of Figures

Figures

List of Tables

Tables

Chapter 1

Introduction

Background

Cerebral microbleeds (CMBs) are small foci of chronic blood products in brain tissues that are critical markers for cerebral amyloid angiopathy (Knudsen et al., 2001). CMBs' presence increases the risk of symptomatic intracerebral hemorrhage and ischemic stroke (Cordonnier, Al-Shahi Salman, & Wardlaw, 2007). CMBs can also cause structural damage to brain tissues resulting in neurologic dysfunction, cognitive impairment, and dementia (Agarwal, Kanekar, & Mittal, 2016).

Advances in magnetic resonance imaging (MRI) technologies, e.g., susceptibility-weighted imaging (SWI) (Akter et al., 2007), make these paramagnetic blood products (hemosiderin) sensitive to screening and facilitating the identification of CMBs (Goos et al., 2011). The traditional method for CMB classification employs human visual rating based on the Microbleed Anatomical Rating Scale (MARS) (Greenberg et al., 2009; Gregoire et al., 2009) that draws up stringent rules to classify CMBs into “definite” and “possible”. Nevertheless, the manual method is both time-consuming and error-prone.

Several semi-automated detection methods have been proposed (Kuijf et al., 2013; Krizhevsky, Sutskever, & Hinton, 2012; Barnes et al., 2011) to improve rating quality and decrease rating time. However, these methods have limitations because of clinical and technical constraints. On the clinical side, significant variation in the size of CMBs (Charidimou et al., 2013); their widely distributed locations (Charidimou & Werring, 2012); and the existence of

CMB mimics, e.g., flow voids, calcification, and cavernous malformations, all of which resemble CMB in SWI scans (Figure 1), may result in CMB misclassification (Greenberg et al., 2009). On the technical side, semi-automated methods came across the following problems: they need intensive preprocessing and prescreening techniques, the detection resolution is rough, and the lack of volumetric data is a problem since the spatial information on all three dimensions is crucial to enhancing CMB detection and classification.

Other deep learning techniques have been proposed to overcome the limitations of the extant machine learning methods on detecting and classifying CMBs with remarkable results. For example, Dou et al. (2016) presented one of these methods using a two-stage, three-dimensional convolutional neural network (3D-CNN) cascade method, achieving 92.31% sensitivity and 42.69% precision.

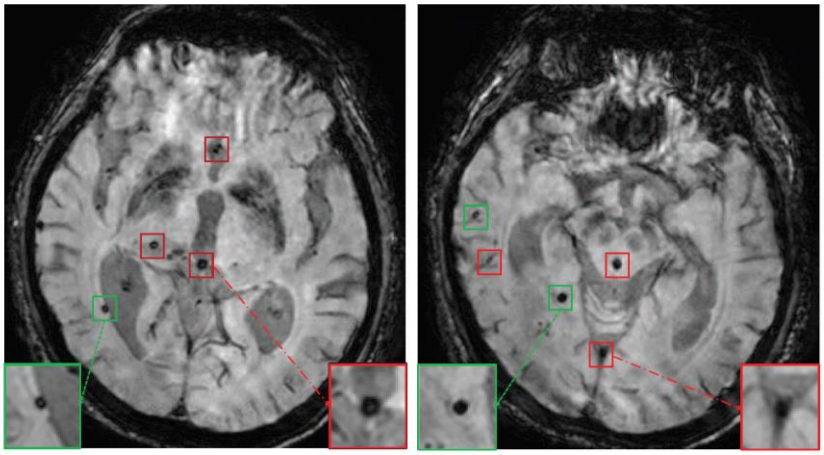


Figure 1 - Examples of true CMB (green rectangles) and CMB mimics (red rectangles).

Problem Statement

The Dou et al. (2016) method for CMB detection and classification consists of *screening* and *discrimination* stages. In the screening stage, a 3D Fully Convolutional Network (3D-FCN) uses MRI volumetric data as its input, outputting 3D score volumes and retrieving potential candidates where the value for each 3D score volume represents the probability of a CMB occurrence at a corresponding voxel of the input volume. Subsequently, a 3D-CNN classifies true CMB from challenging mimics with high-level feature representations in the discrimination stage. The screening stage's 3D-FCN detects lesions using the traditional sliding window strategy (Ciresan, Giusti, Gambardella, & Schmidhuber, 2013; Havaei et al., 2017; Roth et al., 2015) for a volume of 512 ×512 ×150 voxels. However, this arrangement results in the sampling of 39.3 million 3D patches in a voxel-wise manner, resulting in a computational workload that may not be acceptable in clinical practice.

Dou et al. (2016) noted that most convolutional operations are conducted redundantly due to overlapped sampling. The authors enhanced the original CNN algorithm to reduce this overload, significantly removing redundant computations and speeding up the detection process. In addition to the sliding window enhancement, it has been verified that, during the discrimination stage, the screened candidate positions are marked and cropped, constructing a new 3D-CNN model and classifying 3D candidate regions. However, not all CMB are perfectly spherical. Thus the potential CMB volumetric information may not be sufficient to define a voxel volume encompassing all CMB dimensions. The missing volumetric details may cause the model not to identify CMB correctly.

In the CNN architecture, while both convolutional (C) and the max-pooling (M) layers can process arbitrary-sized input, the fully-connected (FC) layers must have a fixed length. The fixed length constraint occurs because the FC inputs require flattening the feature volumes into vectors. Such vectors are generated via a Bag-of-Words (BoW) approach (Yang et al., 2007) that pools the features together, missing both the aspect ratio and scale of the input image (He, Zhang, Ren, & Sun, 2015). This constraint poses additional problems: the cropped region may not contain the entire object, while wrapping may result in unwanted geometric distortion.

To address the FC fixed-size constraint without compromising detection speed, Dou et al. (2016) suggest (without implementation) integrating the Spatial Pyramid Pooling (SPP) strategy (He, Zhang, Ren, & Sun, 2015), allowing the 3D-CNN model to consider multi-scale information during the feature representation phase.

The SPP technique maintains spatial information by pooling local spatial bins with sizes proportional to the image size. Hence, regardless of the image size, the number of bins is predetermined. In each spatial bin, each filter's responses are pooled using max-pooling, and the outputs of the spatial pyramid pooling are *kB*-dimensional vectors, where *B* represents the number of bins, and *k* represents the number of filters in the last convolutional layer. The feature vectors are the *n*-dimensional vectors describing the coordinates of the features of the respective images. The fixed feature-vector output from the previous convolutional layer is the fully-connected layer's input, regardless of the feature maps' size, as shown in Figure 2.

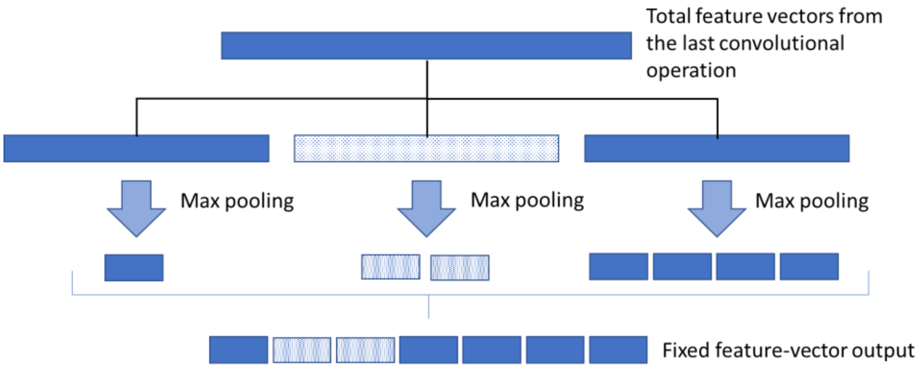

Total feature vectors from the last convolutional operation
Max pooling
Max pooling
Max pooling
Fixed feature-vector output

Figure 2 – SPP layer in feature vector processing (Li et al., 2018; He, Zhang, Ren, & Sun, 2015)

Adding an SPP layer between the last convolutional layer and the following fully-connected layer (Li et al., 2018) removes the model's fixed-size constraint because of the following SPP properties:

1) The SPP allows the CNN to accept inputs of any scale, thus increasing the image scale invariance of the model, suppressing overfitting, and enabling the extraction of local features of the data at multiple levels by generating a fixed-length output regardless of the input size; and
2) The SPP uses multi-level spatial bins while sliding window pooling uses only a single-window format.

The resulting information aggregation avoids the need for cropping images in the discrimination stage and the consequent loss of contextual information, improving CNN's classification accuracy and reducing the training and prediction time.

When combined with a 3D-CNN, SPP can learn features from 3D data such as volumetric images, videos, or point clouds. The basic idea is the same as with 2D-CNN, instead of dividing the input into 2D spatial bins, the 3D data is divided into 3D bins.

Dissertation Goal

This dissertation implements the SPP strategy into the 3D-CNN two-stage model (Dou et al., 2016). It compares the method with the best available techniques for detecting and classifying candidate CMB of different sizes and shapes. As a proof of concept, Dou et al. (2016) 3D-CNN two-stage model is used to baseline the modified 3D-CNN-SPP model to distinguish between two classes of objects: CMB and noise.

As a control, the performance of the modified 3D-CNN model is compared to three classifiers, such as Support Vector Machine (SVM) (Barnes et al., 2011), Random Forest (Liaw & Wiener, 2002), and the classifier used in Dou et al. (2016).

Relevance and Significance

CNNs are feature extractors consisting of convolutional and fully-connected layers that reduce the number of the parameters learned, retaining the most helpful information for a classifier. Recent improvements in the deep CNN's representative capacity have been achieved through architectural innovations (Khan, Sohail, Zahoora, & Qureshi, 2019). However, despite their effectiveness and efficiency, a limitation of the CNNs requires a fixed size input. This constraint arises because fully-connected layers are essentially linear projection operators. In contrast, convolutional layers operate in a sliding window, not requiring fixed-size input to generate feature maps of any size.

Although Dou et al. (2016) method has achieved 92.31% sensitivity, it may misinterpret some actual CMBs due to their volumetric irregularities, resulting in input images of various sizes. A proposed solution for this limitation is integrating the SPP strategy into the 3D-CNN layer between the convolutional and fully-connected layers, determining the optimal pool sizes and vector inputs for the last fully-connected layer.

Therefore, this dissertation's primary contribution is investigating the effects of allowing different-size images to be sent to the same network for training and enhancing the 3D-CNN model classification capabilities on detecting CMB of different sizes and shapes incorporating the SPP. This dissertation's secondary contribution is converting the SPP 2D model into 3D. The original SPP model expects a 2D image input, but when it comes to 3D images, the SPP strategy requires an extra dimension added to the original pooling kernel. Converting the original SPP model into a 3D image pooling operation enables the CNN architecture to accept different image sizes without massive zero-padding effort, reducing noise and improving the model performance.

Summary

This chapter begins by providing background about CMB. Then, it briefly introduces the significance of CMB identification on SWI-MRI for clinical diagnosis and discusses the differences between manual and semi-automated or fully automated classification methods. Next, the chapter defines the goal of the proposed dissertation. Finally, the chapter elaborates on the proposed dissertation's contribution to computer science's knowledge base.

This dissertation is organized as follows: Chapter 2 provides a literature review on semi-automated, traditional machine learning and deep learning methods for CMB detection and classification methods on MRI/SWI. Chapter 3 presents the methodology of the foundation for the investigational portion of the study. Chapter 4 presents implementation details, lists the metrics to be used, runs the experiment, and compares the results. Finally, Chapter 5 concludes the dissertation and provides suggestions for future work.

Chapter 2

Literature Review

Context

CMB identification and classification have been based mainly on human visual identification of MRI/SWI features via shape, size, and intensity information. However, advances in imaging technologies enabled robust computer vision-based systems for automatic CMB detection. In the context of computer vision, Spatial Pyramid Pooling has been considered a suitable solution to improve CNNs classification capability. The following sections review the relevant literature:

- Feature-based approaches
- Machine Learning approaches
- Deep Learning approaches
- Spatial Pyramid Pooling
- Summary

Feature-based approaches

Previous CMB identification and classification investigations have used geometry-based learning features (Lo et al., 1995) to identify and increase nodule detection accuracy on radiographs. The method works by pre-scanning suspect areas and classifying them by a CNN using a modified moment invariance of matrix operations. Despite being developed for radiographs with consequent severe image quality limitations, this method was one of the first attempts to use CNN for semi-automated clinical image nodule detection to achieve 88% accuracy.

Kuijf et al. (2011, 2013) applied fast radial symmetry transform (FRST) (Loy & Zelinski, 2003) to identify spherical regions as CMBs on MR images with 71.2% sensitivity and a detection rate of 50% for total true CMB. While not using machine learning methods for identification and classification, this method's contribution is implementing the 3D FRST algorithm, highlighting the radial symmetry spherical points of interest for CMB detection.

Furthermore, an enhancement of Kuijf's process (Bian et al., 2013) combines 2D FRST detection and 3D region growing to exclude the falsely identified CMBs, achieving 86.5% sensitivity.

Machine Learning approaches

Barnes et al. (2011) present an enhancement in CMB detection using a 3D semi-automatic process and exploiting a Support Vector Machine (SVM) to classify CMBs. This method uses four steps: a preprocessing phase to remove the skull and background features, statistical thresholding to mark all hypointensities, an SVM classifier to eliminate noises and veins, and the manual review of results to remove false positives (FP).

Although the technique has 81.7% sensitivity with an average of 107.5 FPs per subject, its semi-automated process has been considered cumbersome and impractical for clinical use. Therefore, Ghafaryasl et al. (2012a, 2012b) propose a three-step CMB detection and classification framework consisting of preprocessing, classification, and postprocessing phases.

The preprocessing phase consists of two steps: the first one performs soft tissue segmentation. The next task selects CMB potential candidates based on image intensity and size interval. This phase's output is a set of regions of interest (ROI) composed of binary images. The classification phase encompasses feature extraction, selection, and the learning classifier. The feature extraction’s primary goal is to distinguish CMBs from false-positive candidates. The selection phase differentiates blob structures from vessels or plate-like structures. Finally, the classification phase uses the binary Quadratic Discriminant Classifier (QDC) method to distinguish CMBs from veins and noise.

The technique has achieved 98.6% sensitivity, detecting 4.1 false candidates per subject. Despite the method's sensitivity, the authors reported a limitation in finding many false-positives, causing the technique not to be used as a fully automated system. Hence, radiologists have to annotate CMBs manually. The method’s performance has also been considered unsuitable for clinical use.

Fazlollahi et al. (2015) use a multi-scale Laplacian of Gaussian (MSLoG) model to detect the potential CMB candidates and train a cascade of binary random forests. The method improves the detection of spheric objects differentiating CMBs from blood vessels. Nonetheless, a limitation of this solution is the lack of contextual information due to the third dimension's loss.

Dou et al. (2015a) propose a three-stage approach where the first phase performs CMBs candidate screening based on intensity. The second stage performs a 3D hierarchical feature extraction via a stacked convolutional Independent Subspace Analysis (ISA) network. Finally, the third stage removes false-positive candidates with an SVM classifier based on the learned representation features from ISA. The method achieved 89.44% sensitivity, averaging 7.70 false positives per subject and 0.90 false positives per total CMBs.

Ateeq et al. (2018) proposed an ensemble classifier-based system consisting of three steps: first, the image of the brain is extracted from surrounding rigid structures, then the initial candidates are identified on the threshold layer; finally, the third step performs the feature extraction and classification of CMBs from other healthy tissues to remove FPs using an SVM, Quadratic Discriminant Analysis (QDA) and ensemble classifiers.

Deep Learning approaches

One main limitation of previous methods is considering low-level features only, like intensity and geometry (shape, compactness, and size) for CMB detection, requiring a post-manual review process to eliminate the large number of FPs. To overcome such limitations, Ciresan, Giusti, Gambardella, & Schmidhuber (2013) and Roth et al. (2015) proposed using single-slice images for data collection, processing them sequentially on a 2D-CNN.

Chen et al. (2016) presented a complementary approach to this method using the aggregation of adjacent slices or orthogonal planes to provide additional spatial information.

In another work, Dou et al. (2015b) proposed a three-phased pipeline for CMB automatic detection via deep learning-based 3D feature representation. In the first phase, the CMB candidates are located by statistical thresholding. The second phase uses a deep convolutional neural network (CNN) for hierarchical 3D feature representation. In the last step, an SVM classifier is trained to distinguish between true CMB and CMB mimics, achieving 89.13% sensitivity with an average of 6.4 FPs per subject.

Zhang et al. (2016) used a Sparse Autoencoder (SAE) for unsupervised feature learning for voxel-wise detection (Tao & Cloutie, 2018) with a four-layered deep neural network achieving 93.2% sensitivity. In a subsequent study, Zhang et al. (2017) compared the results of a single hidden layer feed-forward neural network using three different activation functions: logistic sigmoid (LOSI), rectifier linear unit (ReLU), and leaky rectified unit (LReLU), achieving 93.05% sensitivity with LReLU as the best result.

Recently, a seven-layer deep neural network (DNN) model based on the sparse autoencoder and softmax classifier (Zhang et al., 2018) for voxel detection CMBs achieved 95.13% sensitivity, 93.33% specificity, and 94.23% accuracy.

Despite their accuracy and efficiency, the previous solutions could still not fully use volumetric spatial information because the input slices are treated independently. Therefore, Dou et al. (2016) propose a novel approach using 3D convolutional kernels as a more reliable solution that takes full advantage of CMBs' spatial information. In addition, the authors introduced a method for identifying CMB candidates with higher sensitivity, performing a fine-grained discrimination selection, and avoiding redundant computations used in the traditional sliding window technique.

Detecting lesions using the conventional 3D CNN sliding window method has been demonstrated to be impractical, especially for high-resolution image inputs, because thousands or millions of 3D block samples must be analyzed. The strategy proposes a workaround consisting of two cascaded stages, *screening* and *discrimination*. In the screening stage, the 3D FCN retrieves potential candidates, while in the discrimination stage, the 3D CNN focuses on these candidates to accurately single out the real CMBs from challenging mimics. The screening stage includes the *training* and *testing* phases.

In the *training* phase, the positive samples are extracted from candidate regions. The network is then trained in three sub-steps. The first step consists of initial 3D-CNN training with randomly selected non-CMB regions throughout the brain (negative samples). Next, false-positive samples acquired via the initial model are added to the training set. The initial model is then fine-tuned, and regions with high prediction probabilities are selected as potential candidates.

The 3D-FCN model inputs the whole volume and generates the corresponding coarse 3D score volume during testing. Since the produced score volume could be noisy, the model utilizes local non-max suppression in post-processing. According to the index mapping process proposed, the 3D score volume locations are then traced back to coordinates in the original input space. Finally, a newly constructed 3D-CNN model classifies the extracted 3D candidate regions in the discrimination stage.

To generate representative samples and improve the 3D-CNN model's discrimination capability, the FPs from the screening stage's training set are used as negative samples when training the 3D-CNN in the second stage. Thus, the 3D-CNN captures the spatial contextual information and hierarchically extracts high-level features from the 3D images. As a result, it performs better than previous methods based on low-level 3D features or 2D-CNN.

Spatial Pyramid Pooling

The traditional CNN architecture consists primarily of convolutional (C) and fully-connected (FC) layers. The convolutional layers (C) slide a kernel over the image, and the number of features is determined by the number of different kernels used to slide over the input. i.e., K different kernels will result in W different feature maps. The fully connected (FC) layers take the convolution/pooling process outputs classifying the image.

As previously mentioned, a limitation of the CNN is the input (in this case, an image) fixed size requirement. The sliding kernel strategy used in the convolutional layers can cope with any image size, but the fully connected layers have a fixed-sized input by construction. Therefore, the fully connected layers impose the size restriction, and input images often must be cropped or warped to fit the network's size requirements.

To overcome the fixed-size input limitation, He, Zhang, Ren, & Sun (2015) proposed a suitable solution based on a kernel-based image categorization (Lazebnik, Schmid, & Ponce, 2006) by adopting an efficient approximation technique from the pyramid matching scheme (Grauman & Darrell, 2005) called Spatial Pyramid Pooling (SPP).

The method, shown in Figure 3, consists of implementing an SPP layer on top of the first fully connected layer after the last convolutional layer to avoid the need for the image to be resized, i.e., the input image can have any size (Qu, Zhang, & Sun, 2017). Furthermore, this arrangement allows different-sized pictures to be sent to the same network for training, avoiding complexity during feature extraction and classification (Li et al., 2018) and mapping any size input down to a fixed size output.

The SPP divides the last convolutional layer feature maps output into uniform sub-regions or bins at different resolution levels and then applies a pooling operator on these sub-regions separately. Therefore, the number of bins is fixed regardless of the image size.

The bins dimensions are proportional to the image, with shapes $c \times 4 \times 4$, $c \times 2 \times 2$, and $c \times 1 \times 1$, for example, where c is the number of filters of the convolutional layer before the SPP as shown in Figure 3. Each filter's response is pooled using max-pooling. The final representation is obtained by concatenating different sub-regions features, resulting in a fixed feature-vector output with the fully connected layer's input.

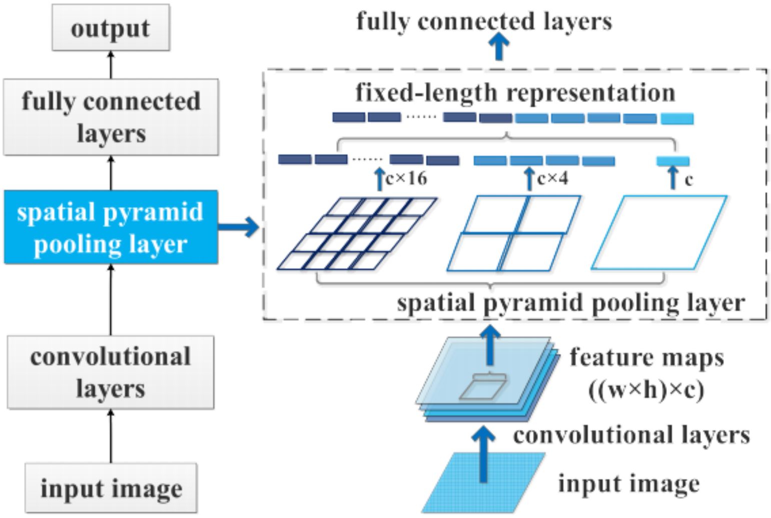


Figure 3 – SPP layer architecture (He, Zhang, Ren, & Sun, 2015)

Summary

This chapter reviewed extant CMB identification and classification investigations using geometry and intensity features. The chapter continued by examining traditional machine learning methods for detection and classification. The last section reviews the relevant literature and SPP architecture concepts to be implemented in this work. The next chapter addresses the proposed methodology for the dissertation.

Chapter 3

Methodology

Overview

This dissertation investigates incorporating SPP into the original Dou et al. (2016) 3D-CNN model for CMB detection and classification. This chapter describes the study's approach in the following sections:

1) Description of the experiment dataset.
2) Reproducing the previous results of the SVM, Random Forest, and the original two-stage 3D-CNN baselining the model.
3) Incorporating the SPP into the 3D-CNN two-stage model.

Description of the experiment dataset

A three-dimensional (3D) brain MRI comprises a sequence of 2D magnetic resolution images generally of the same height, width, and thickness greater than zero, as shown in Figure 4. Therefore, the image is composed of voxels instead of pixels. If voxels are equal in size in all three dimensions (e.g., 3mm × 3mm × 3mm), they are considered *isotropic,* and the voxel size is the spatial resolution of the image (Tariq & Burney, 2014).

In the MRI-SWI, CMBs appear as small black circles because they are deposits of paramagnetic blood products (hemosiderin) (Haake et al., 2009). The hemosiderin absorbs the magnetic fields of the MRI, causing a hypointense image.

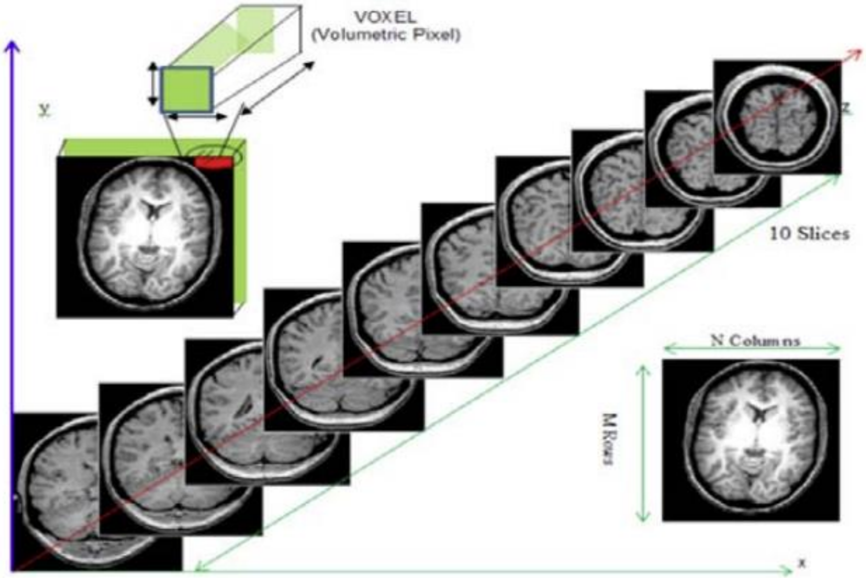


Figure 4 - Image elements of brain 3D MRI with ten slices (Tariq & Burney, 2014).

The Dou et al. (2016) study used 320 MRI-SWI images acquired from a 3.0T scan with a volume size of $512 \times 512 \times 150$ voxels, slice thickness of 2mm, slice spacing of 1mm, and a 230 x230mm^2 field of view. Hence, each sample is an array size of $512 \ \times 512 \times 150$, i.e., 39,321,600 voxels.

The dataset was labeled by an experienced rater and verified by a neurologist following the Microbleed Anatomical Rating Scale (Greenberg et al., 2009; Gregoire et al., 2009). In addition, a set of 20 SWI images (out of 320) has been made available by Dou et al. (2016). The 20 example files containing 1,149 labeled CMBs are available at http://www.cse.cuhk.edu.hk/~qdou/cmb-3D-CNN/cmb-3D-CNN.html. All models developed use the twenty images subset to train the model.

Figure 5 shows an example MRI-SWI image. The ground truths are the three cartesian coordinates of annotated CMB.

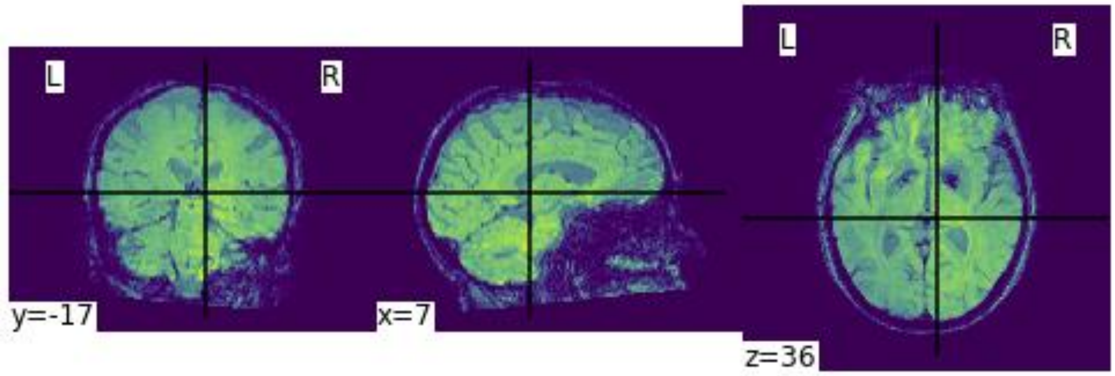


Figure 5 – MRI-SWI image (01.*nii*)

The following section aims to replicate the previous works of Barnes et al. (2011) (SVM), Liaw & Wiener (2002) (Random Forest), and Dou et al. (2016) 3D-CNN. These models are baseline models compared to the developed 3D-CNN-SPP model.

Reproducing SVM and Random Forest

SVM (Barnes et al., 2011) and Random Forest (Liaw & Wiener, 2002) classify the CMBs based on geometry features. For example, a cluster of adjacent voxels with a common centroid forms a Region of Interest (ROI) (Figure 7).

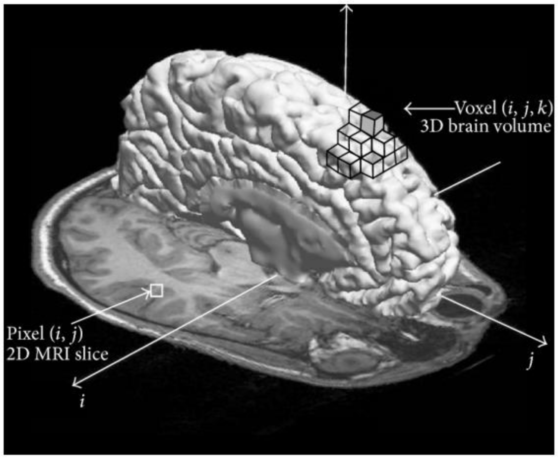


Figure 7 – Voxel cluster (ROI) (Haake et al., 2009)

Each ROI has a seed voxel, which determines the candidate neighbor voxels based on their intensities (Newman et al., 1996), as shown in Figure 8.

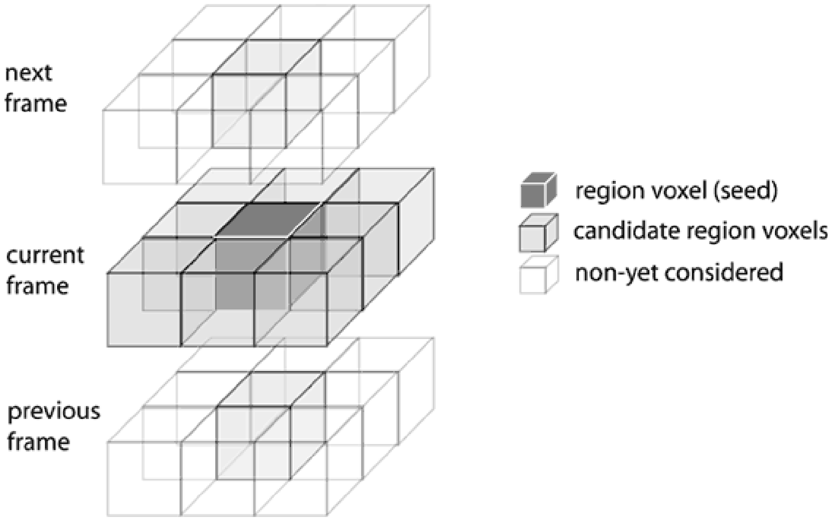


Figure 8 – ROI candidate voxels (Newman et al., 1996)

The selected voxels within an ROI define a volume (Figure 9) that has its geometric features — volume (*V*), area (*A*), region compactness (C), and fractional anisotropy (FA).

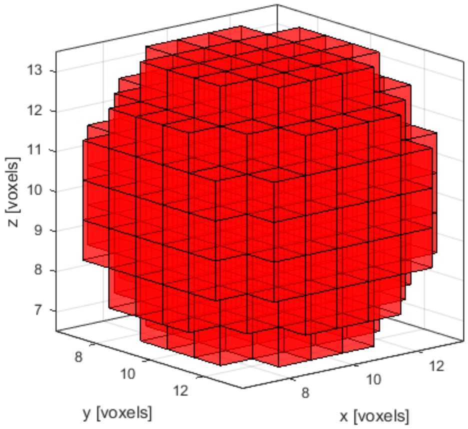
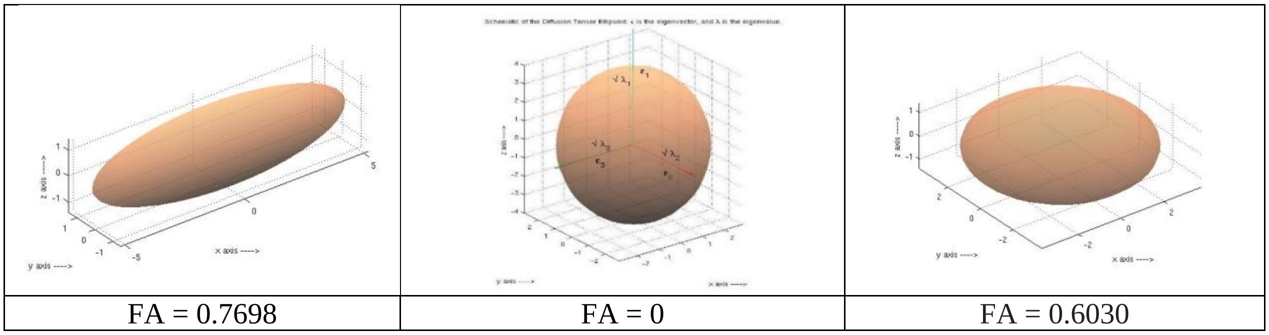


Figure 9 – 3D plot of voxels in a binary matrix

Because CMBs have a characteristic spherical shape, Fractional Anisotropy (FA) helps differentiate between CMB and veins. FA values near zero (0) indicate a sphere. On the other hand, values above 0.7 may indicate veins or other tubular structures (Cohen-Adad et al., 2008), as shown in Figure 10.


Schematic of the Diffusion Tensor Ellipsoid: ε is the eigenvector, and λ is the eigenvalue.
z axis ---->
y axis ---->
x axis ---->
√λ1
ε1
√λ2
ε2
√λ3
ε3

FA = 0.7698	FA = 0	FA = 0.6030

Figure 10 – Fractional Anisotropy values (Cohen-Adad et al., 2008)

Dou et al. (2016) provide a candidate ROI list for each image. The selected dataset contains 317 ROI centroids preprocessed to calculate the ROI geometry (Compactness and Fractional Anisotropy) using the Python library PyRadiomcs (Griethuysen et al., 2017).

The SVM and Random Forest were trained with 105 samples (33%) of the dataset. The SVM model hyperparameters are kernel = rbf (default), a training set size = 33%, and regularization parameter C = 0.08.

The resulting confusion matrix (Figure 11) is consistent with the annotated training dataset (105 candidates with five confirmed CMBs) with 86.67% accuracy, 100% precision, and 26.32% recall for CMB (class = 1).

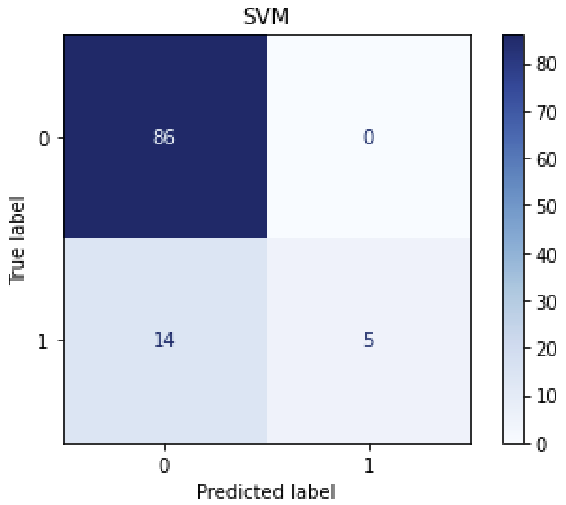


Figure 11 – SVM classifier confusion matrix

The Random Forest model used 500 trees, with a maximum depth of ten levels for each tree, resulting in 90.47% accuracy, 76.47% recall, and 68.42% precision.

Figure 12 shows the confusion matrix, and Figure 13 shows both models' Receiver Operating Characteristic (ROC) curves.

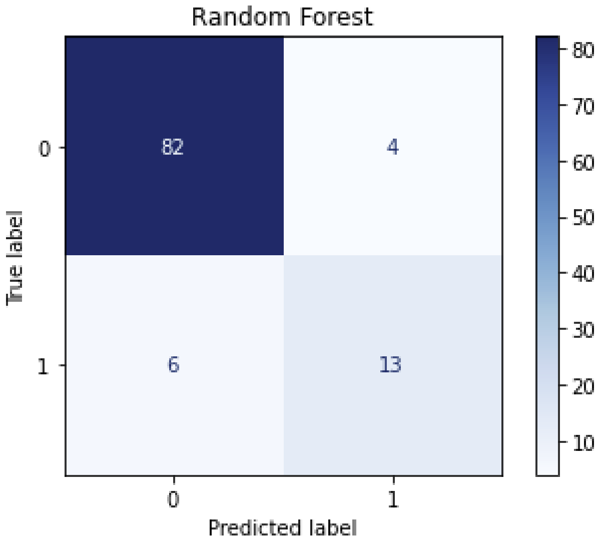


Figure 12 – Random Forest classifier confusion matrix

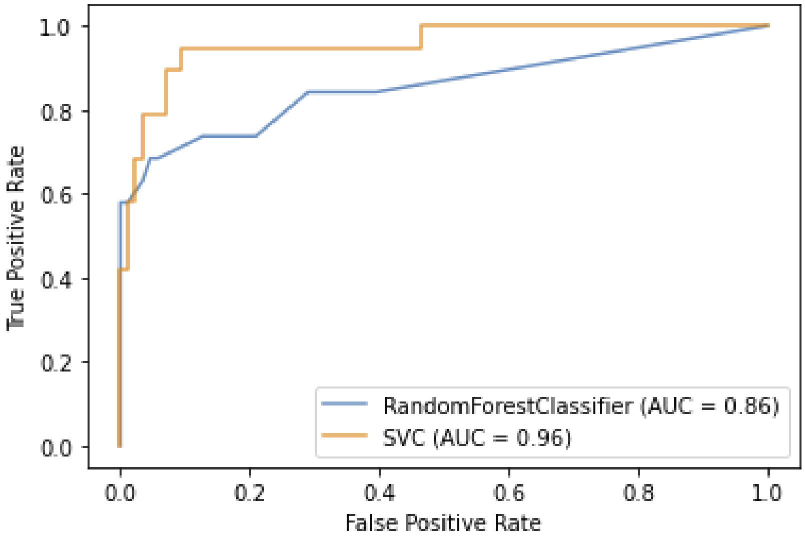


Figure 13 – Receiver Operating Characteristic (ROC) curve

Comparing confusion matrices and the ROC chart curves, the results are consistent with Dou et al. (2016) findings, baselining the model.

The 3D-CNN cascaded model

As mentioned, the Dou et al. (2016) framework is based on a cascaded architecture composed of a *screening and discrimination* stage, as shown in Figure 14.

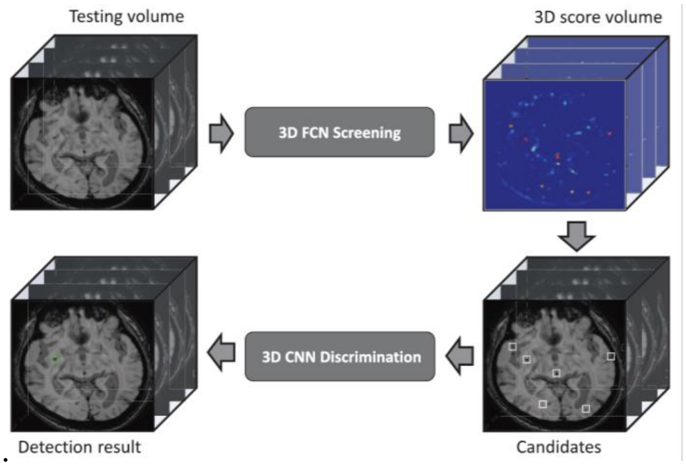


Figure 14 – The cascaded framework for CMB detection (Dou et al., 2016)

In the *Screening* stage, the 3D-FCN model retrieves the SWI image volume as input and outputs a 3D prediction score volume. Each value on the 3D score volume traces back to a candidate Region of Interest (ROI) in the input volume. In this stage, Dou et al. (2016) developed a novel sliding window strategy that avoids redundant computations, improving the 3D-FCN and dramatically accelerating the detection process.

The *Screening* stage consists of two steps *training* and *testing* (Figure 15). During the training phase, the goal is to extract and augment the candidate CMB regions to expand the training database.

The *training* phase consists of three sub-steps. The first step (1) trains an initial traditional 3D-CNN with positive samples and randomly generates negative samples throughout the brain image. The second step (2) augments data, adding false-positive samples and applying the initial model to the training set. Finally, the third step (3) fine-tunes the initial 3D-CNN with the enlarged training database.

In the *testing* phase, the fine-tuned 3D-CNN is converted into the 3D-FCN model by transforming the FC layers into a shape consistent with the convolution layer. Therefore, the 3D-FCN takes the whole volume as input and generates the coarse 3D score volume.

Finally, in the *Discrimination* stage, the 3D-CNN model encodes spatial contextual information to extract high-level features from candidate CMB, distinguishing positive ones and removing false positives.

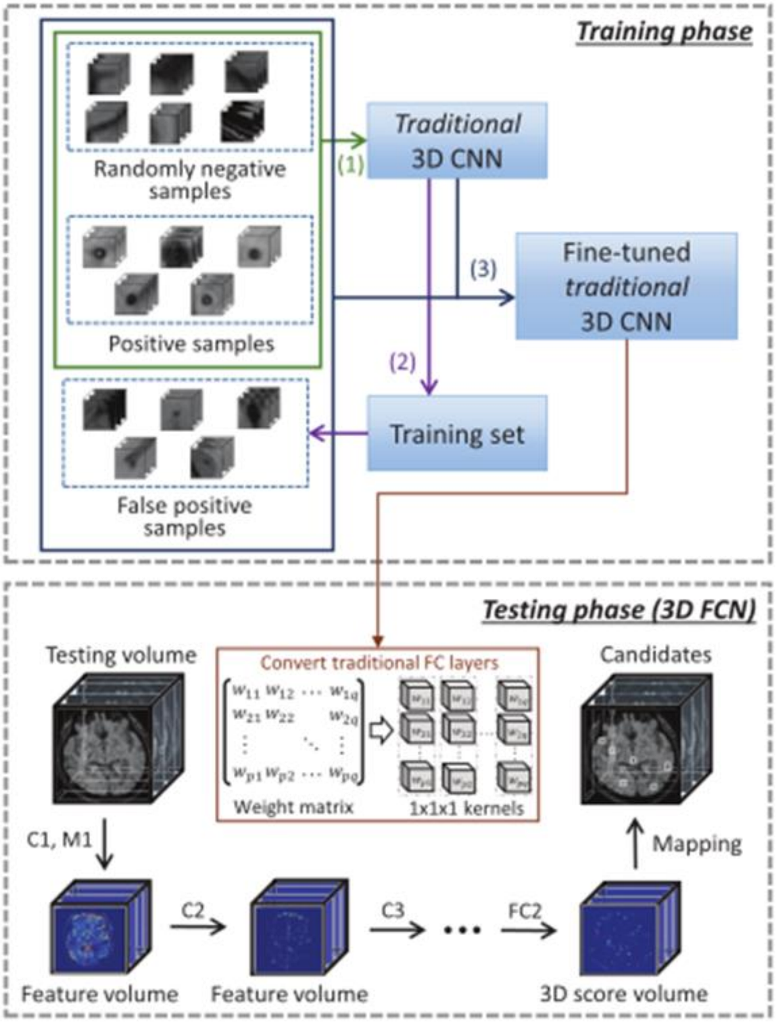


Figure 15 – The *Screening* stage of CMB cascaded framework (Dou et al., 2016)

Reproducing the 3D-CNN cascaded model

The original version of the Dou et al. (2016) code uses an outdated Python version and a deprecated library (Theano), making it unsuitable to work with the most recent NVIDIA® GPU CUDA® drivers and Python 3.19. Hence, it was necessary to entirely rewrite the code using modern ML frameworks, such as Keras (Charles, 2013) and TensorFlow® (Abadi et al., 2016).

In practical terms, the *training* phase starts by preprocessing the images of the testing volume, reading the positive CMBs labeled by specialists from the ground truth files (.*nii*), and saving the candidate centroids of positive CMBs (“***Positive samples***” in Figure 15). The next step creates the random negative samples (“***Randomly negative samp*****les**” in Figure 15), creating a random centroids list.

The positive and negative sample lists are combined as the input for the traditional 3D CNN (Table 1), generating a list of false-positive samples and enlarging the training database. The input size 16x16x10 is selected because it bounds most CMB lesions.

The traditional 3D CNN is then fine-tuned with the enlarged database to strengthen its discrimination capability using the same model as Table 1.

Layer	Kernel	Stride	Output size	Feature volumes
Input	-	-	16x16x10	1
C1	10x10x7	1	7x7x4	32
M1	1x1x1	1	7x7x4	32
C2	6x6x2	1	2x2x3	64
FC1	-	**-**	-	500
FC2	-	**-**	-	100
FC3	-	**-**	-	2

Table 1 – 3D-CNN screening model

Once the *training* phase is done, the resulting fine-tuned 3D-CNN model is saved and used as the input for the *testing* phase. Finally, the traditional FC layers are converted into a shape consistent with the convolution layer (Figure 15 - brown box).

During the *testing* phase, the 3D-FCN takes a whole volume as input, extracts representative feature volumes, and produces a coarse 3D score volume to retrieve candidates according to the architecture of Table 2.

Layer	Kernel	Stride	Output size	Feature volumes
Input	-	-	16x16x10	1
C1	5x5x3	1	12x12x8	64
M1	2x2x2	2	6x6x6	64
C2	3x3x3	1	4x4x2	64
C3	3x3x1	1	2x2x2	64
FC1	2x2x2	1	1x1x1	150
FC2	1x1x1	1	1x1x1	2

Note: FC Layers converted into convolutional layers

Table 2 – 3D-FCN Screening Model

After the 3D-FCN process finishes, the image sections are bundled in one package resulting in an image with the expected dimensions. Next, the potential CMBs' resulting scores are stored in a MATLAB® file and post-processed to generate the possible CMB candidates. Figure 16 shows a map of the listed candidates.

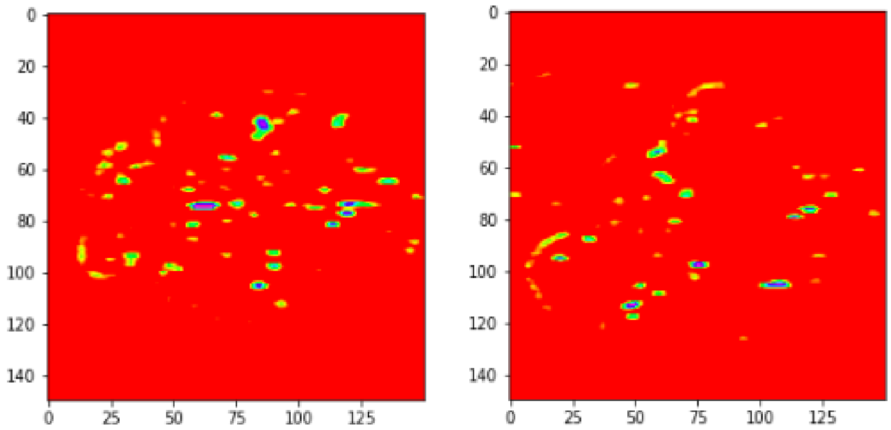


Figure 16 - 3D FCN results (Left - X-axis, Right - Y-axis)

The *Discrimination* stage classifies the true CMBs with high sensitivity and low false-positive rate obtained from the previous stage using the 3D-CNN model in Table 3. For this purpose, Dou et al. (2016) evaluated different receptive field (input) sizes, finding the optimal input size of 20×20×16. The classified CMBs are stored in a MATLAB® format file resulting mask file. The file is post-processed and linked to the raw images' centroids, as in Figure 17.

Layer	Kernel	Stride	Output size	Feature volumes
Input	-	-	20x20x16	1
C1	7x7x5	1	14x14x12	32
M1	2x2x2	2	7x7x6	32
C2	5x5x3	1	3x3x4	64
FC1	-	-	-	500
FC2	-	-	-	100
FC3	-	-	-	2

Table 3 – 3D CNN Discrimination Model

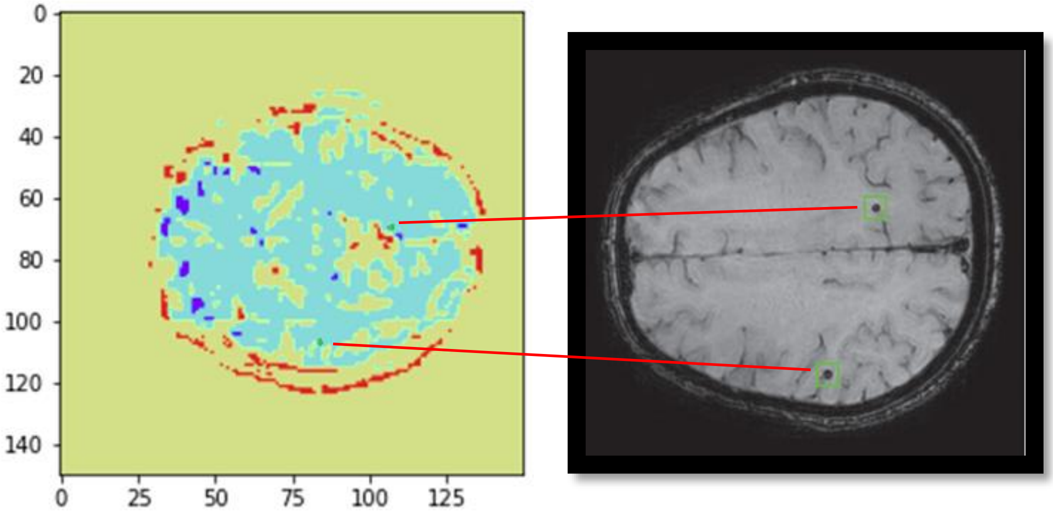


Figure 17 – Resulting mask linked to raw image marked with positive CMB (green boxes)

Incorporating the SPP into the 3D-CNN two-stage model

As mentioned in Section 2, in 2015, He, Zhang, Ren, & Sun equipped a network with a spatial pyramid pooling to eliminate the fixed-size input constraint imposed by the fully connected layers. Spatial pyramid pooling technology generates a fixed-length vector output due to its use of multi-level spatial bins, regardless of the input size. Hence, Dou et al.(2016)

suggested incorporating the SPP to handle outlier CMBs that might be missed, incorrectly detected, or misclassified.

Incorporating the SPP into the 3D CNN model requires adapting the SPP technique to a 3D model proposed by Rao, Tang, & Zhang (2020). The adaptive SPP layer input values are the sides of each cubic spatial bin and the previous convolutional or max-pooling layer features size. For example, a layer of 3 bins of 1, 2, and 4 sides will result in a $1^3 + 2^3 + 4^3 = 73$ feature map. Each bin is multiplied by the previous layer output size (*c*), as shown in Figure 18. In the example, the last convolutional layer output is 64, i.e., $(1^3 * 64 + 2^3 * 64 + 4^3 * 64) = 4{,}672$, the size of the fixed-length resulting vector.

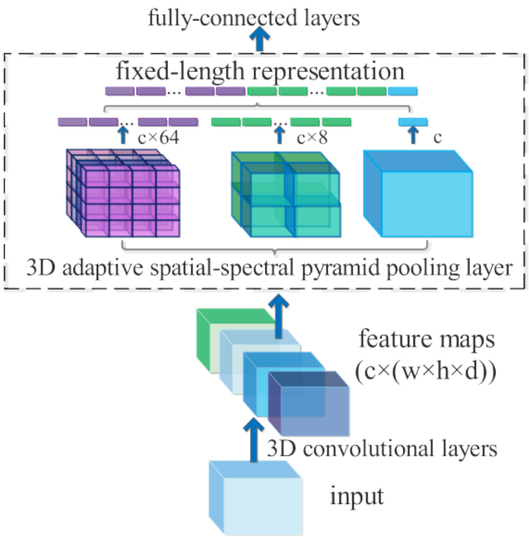


Figure 18 – 3D adaptive spatial pyramid pooling (3D SPP) (Rao, Tang, & Zhang, 2020)

The 3D adaptive spatial pyramid pooling algorithm **`SpatialPyramidPooling`** input parameters (`shape[0], shape[1], shape[2]`) represent the dimension of each spatial bin, and `size` is the previous layer feature volume output. The class output is a vector containing the input vector for the next fully connected layer.

Algorithm 1 3D Spatial Pyramid Pooling

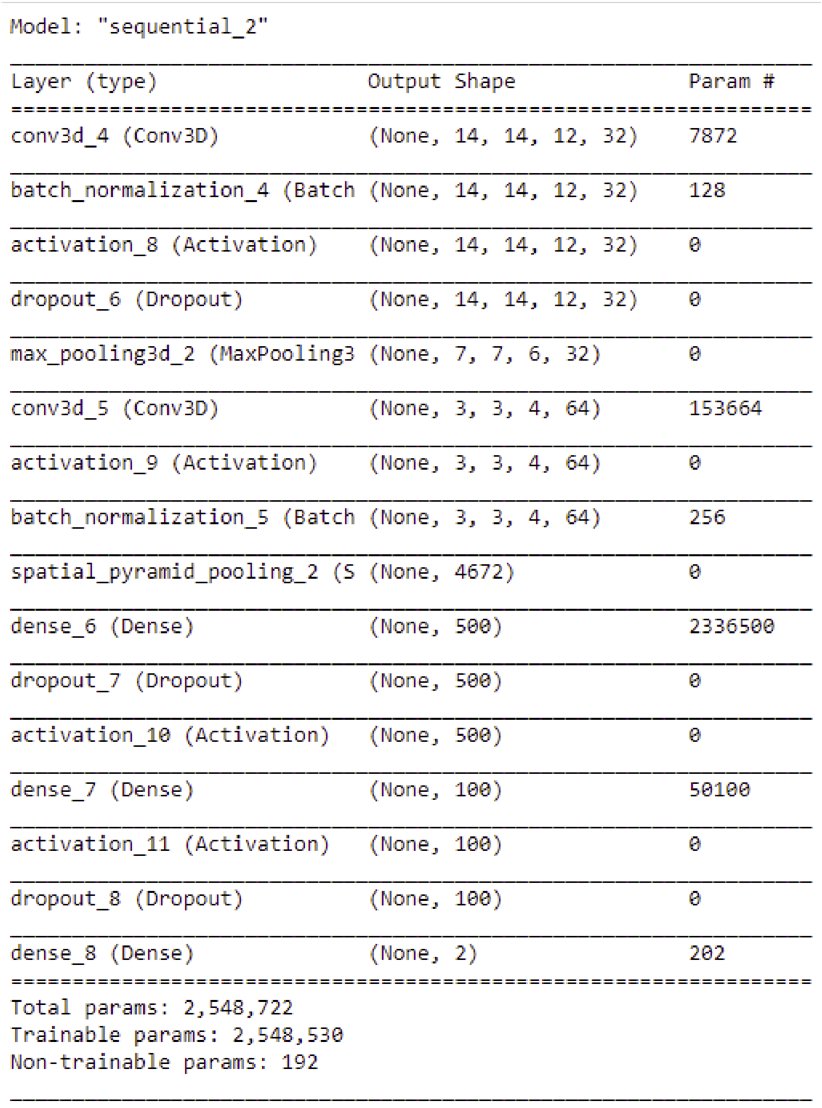
Input: shape[0],shape[1],shape[2],size **Output**: []

```
def: row_length[], col_length[], depth_length[], outputs[]
def: num_pool_regions, pool_num, pooled_val, output
def: x1, y1, z1, x2, y2, z2, ix, jy, kz
num_pool_regions ← pool list length - 1
for pool_num ← 0 to num_pool_regions do
    col_length[] ← shape[pool_num]
    row_length[] ← shape[pool_num]
    depth_length[] ← shape[pool_num]
for pool_num ← 0 to num_pool_regions do
  for kz ← 0 to num_pool_regions do
    for jy ← 0 to num_pool_regions do
      for ix ← 0 to num_pool_regions do
          x1 ← ix * col_length[pool_num]
          x2 ← ix * col_length[pool_num] + col_length[pool_num]
          y1 ← jy * row_length[pool_num]
          y2 ← jy * row_length[pool_num] + row_length[pool_num]
          z1 ← kz * depth_length[pool_num]
          z2 ← kz * depth_length[pool_num] + depth_length[pool_num]
          new_shape = (shape[0], z2 - z1, y2 - y1, x1 – x2, shape[4])
          pooled_val ← max(new_shape)
          outputs ← pooled_val
output ← concatenate(outputs)
return output
```

Table 4 shows the 3D-CNN model with the 3D adaptive SPP layer. The implemented model is shown in Figure 19.

Layer	Kernel	Stride	Output size	Feature volumes
Input	-	-	20x20x16	1
C1	7x7x5	1	14x14x12	32
M1	2x2x2	2	7x7x6	32
C2	5x5x3	1	3x3x4	64
SPP	(1,2,4)	-	-	4672
FC1	-	-	-	500
FC2	-	-	-	100
FC3	-	-	-	2

Table 4 – 3D-CNN with SPP layer



```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv3d_4 (Conv3D)            (None, 14, 14, 12, 32)    7872      
_________________________________________________________________
batch_normalization_4 (Batch (None, 14, 14, 12, 32)    128       
_________________________________________________________________
activation_8 (Activation)    (None, 14, 14, 12, 32)    0         
_________________________________________________________________
dropout_6 (Dropout)          (None, 14, 14, 12, 32)    0         
_________________________________________________________________
max_pooling3d_2 (MaxPooling3 (None, 7, 7, 6, 32)       0         
_________________________________________________________________
conv3d_5 (Conv3D)            (None, 3, 3, 4, 64)       153664    
_________________________________________________________________
activation_9 (Activation)    (None, 3, 3, 4, 64)       0         
_________________________________________________________________
batch_normalization_5 (Batch (None, 3, 3, 4, 64)       256       
_________________________________________________________________
spatial_pyramid_pooling_2 (S (None, 4672)              0         
_________________________________________________________________
dense_6 (Dense)              (None, 500)               2336500   
_________________________________________________________________
dropout_7 (Dropout)          (None, 500)               0         
_________________________________________________________________
activation_10 (Activation)   (None, 500)               0         
_________________________________________________________________
dense_7 (Dense)              (None, 100)               50100     
_________________________________________________________________
activation_11 (Activation)   (None, 100)               0         
_________________________________________________________________
dropout_8 (Dropout)          (None, 100)               0         
_________________________________________________________________
dense_8 (Dense)              (None, 2)                 202       
=================================================================
Total params: 2,548,722
Trainable params: 2,548,530
Non-trainable params: 192
_________________________________________________________________
```

Figure 19 – 3D-CNN with the SPP layer

Summary

This chapter described the experiment dataset, the SVM and Random Forest results classifying CMB, and the original Dou et al. (2016) two-stage cascade 3D-CNN model. Moreover, it expands the SPP architecture, how SPP translates into a three-dimensional model, how to incorporate SPP into the 3D-CNN two-stage mode, and the resulting 3D-CNN-SPP model. Chapter 4 presents details of the used dataset, implementation, hyperparameters selection discussion, and test results.

Chapter 4

Model Implementation and Results

Introduction

This chapter details the used dataset, the model implementation, and hyperparameters and compares the performance of the 3D-CNN-SPP model against the updated Dou et al. (2016) model. The accuracy of the classifications is evaluated using the same metrics used by Dou et al. (2016) to score sensitivity and precision. Additionally, it discusses using the true-positive (TP) metric as a more relevant parameter for CMB diagnostic.

Dataset Details

The dataset comprises 20 brain SWI volumes made available by Dou et al. (2016), denoted as *SWI-CMB*. The volumes were acquired from a 3.0T Philips Medical System with 3D spoiled gradient-echo sequence with a repetition time of 17ms, echo time of 24ms, volume size 512 × 512 × 150, in-plane resolution 0.45 × 0.45mm, slice thickness 2mm, slice spacing 1mm, and 230 × 230mm^2 field of view. The resulting *SWI-CMB* is composed of 148 images with 178 annotated CMBs and 418 candidate CMBs from the FCN (Table 5).

	Images	Annotated CMBs (Ground truth)	Candidates (from FCN)
Training	60	53	138
Testing	88	125	280
Total	148	178	418

Table 5 – *SWI-CMB* Dataset

Implementation and Hyperparameters Selection

As seen before, the suggested model replaces the 3D-CNN *flattening* layer before the FCN with an SPP layer. Changing the model architecture implies reviewing the network parameters and reevaluating network performance. For example, the learning and regularization rates significantly impact the network's adaptability and performance (Santos & Papa, 2022).

On the other hand, deep neural networks are prone to overfitting because of the considerable number of parameters to learn (Sabiri et al., 2022). One way to address overfitting is to use dropout or weight decay techniques. Another approach is augmenting the training data by applying random transformations to the images, such as rotation, scaling, and flipping. However, since the SPP layer's purpose is to handle images of different sizes, there is no need for cropping, circumventing data augmentation. Implementing the capability to handle original image sizes can help the network to learn more robust features that are less dependent on the specific position or orientation of the object in the image.

Lastly, adding L1 or L2 regularization terms to the loss function can prevent overfitting by penalizing large weights and promoting more generalized solutions (Kolarik et al., 2020).

Since the used dataset is small (n = 418), the model tends to overfit. Therefore, it uses L2 regression, dropout, early stopping, and small batch size (batch = 2) to prevent the network not converging. As a side note, testing larger batch sizes (batch >1000) yields inconclusive results, while small-batch sizes provide more up-to-date calculations, producing more stable and reliable training, as demonstrated by Masters & Luschi (2018).

Selected Metrics

This phase aims to run the 3D-CNN-SPP model against Dou et al. (2016) and compare the results. The metrics used to score the performances quantitatively are accuracy, loss, sensitivity, precision, recall, f1, number of false positives, and number of true positives. Sensitivity (S), precision (P), and the average number of false positives per subject (FP_{avg}) are defined as follows:

$$S = \frac{TP}{TP+FN}, P = \frac{TP}{TP+FP}, FP_{avg} = \frac{FP}{N},$$

TP, FP, and FN denote the total number of true-positive, false-positive, and false-negative detection results. Here, *N* represents the number of subjects in the testing dataset.

The f1 score is the harmonic mean of precision and recall scores. When the value of f1 is high, the precision and recall are high. A lower f1 score means a more significant imbalance between precision and recall. The precision-recall curve shows the tradeoff between the precision and recall values for different thresholds.

$$f1 = 2 \frac{Precision\ Score\ Recall\ Score}{(Precision\ Score + Recall\ Score)}$$

The 3D-CNN updated model

The original Dou et al. (2016) model has been reproduced, baselining the experiment. The updated model outperformed the original (Table 6). Nonetheless, the comparison results are close enough (< 1% variance).

Method	Sensitivity	Precision	FP_{avg}
3D-CNN (Dou et al., 2016) (Original)	92.31%	42.69%	2.90
3D-CNN (Updated)	94.45%	45.69%	2.32

Table 6 – Original x Updated 3D-CNN Performance

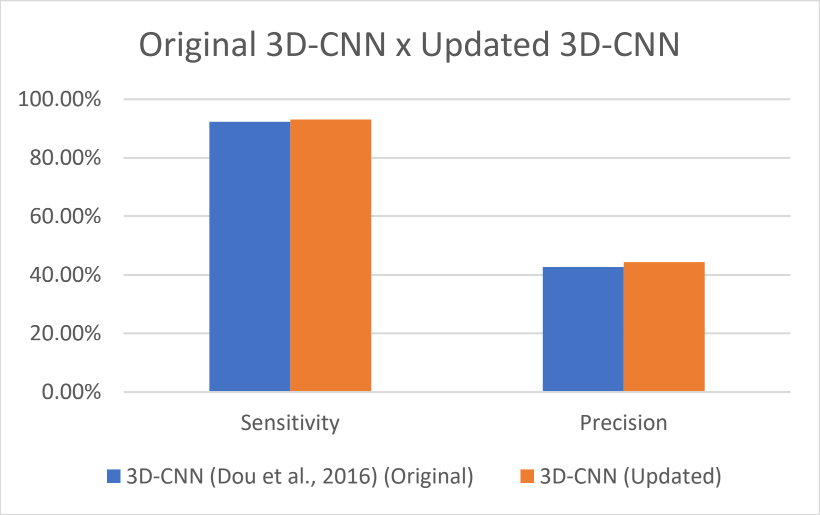
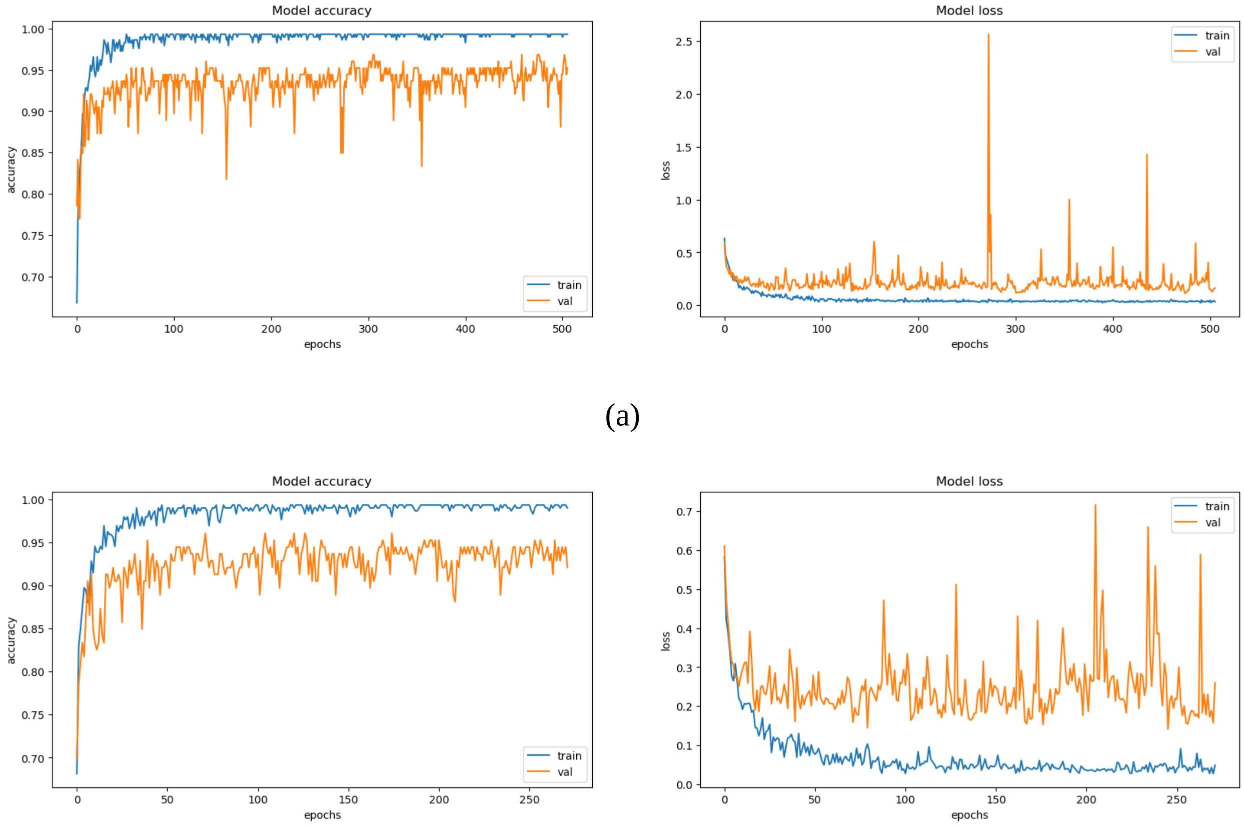


Figure 20 – Original x Updated 3D-CNN Performance

Experiment and Comparison with state-of-the-art methods

Running the 3D-CNN-SPP and 3D-CNN models, the binary cross-entropy (Figure 21) shows that the model performed well, achieving 98.28% accuracy on the training and 96.41% on the test datasets with an f1 score of 96.20%. Figure 22 compares 3D-CNN-SPP and 3D-CNN after five runs. Table 7 shows that 3D-CNN-SPP outperformed the 3D-CNN by 2.16%.



(a)

(b)

Figure 21 – 3D CNN – Binary Cross-entropy – (a) Without SPP and (b) With SPP

	Accuracy		Loss	
Method	Train	Test	Train	Test
3D-CNN (Dou et al., 2016)	98.15%	94.45%	0.1814	0.4316
3D-CNN-SPP	**98.28%**	**96.41%**	**0.1473**	**0.3922**

Table 7 – Accuracy & Loss – 3D-CNN x 3D-CNN-SPP

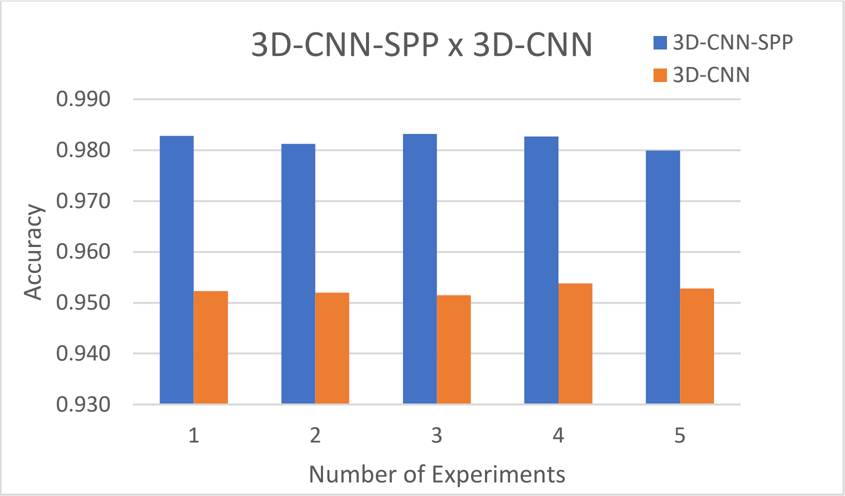
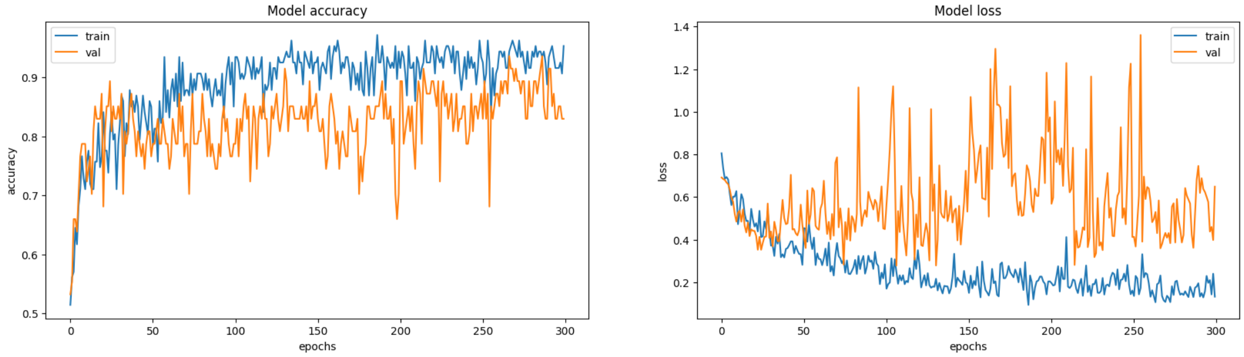
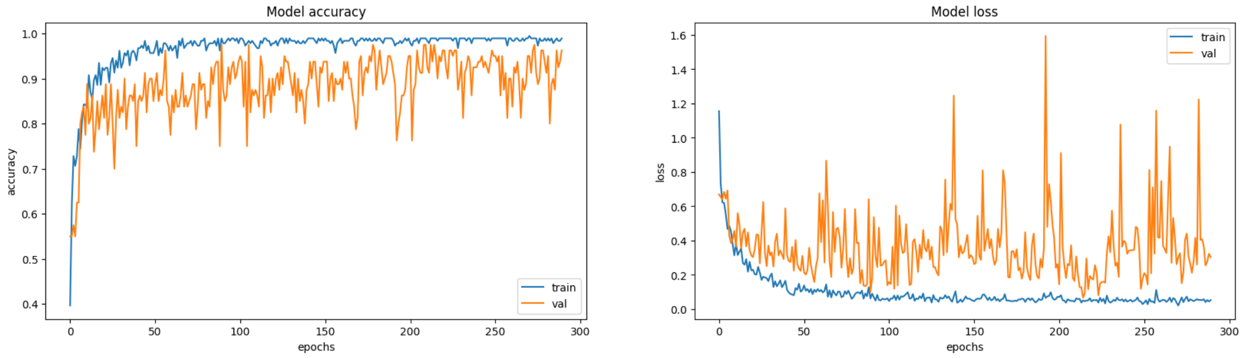


Figure 22 – 3D-CNN-SPP x 3D-CNN performance

For recall evaluation, the experiment determined how the used threshold affects the results, deciding whether the detection is positive. Figure 23 (a) and (b) and Table 8 show 3D-CNN-SPP model behavior using different input sizes, 16×16×10 and 20×20×16, and how the SPP layer handled different scales.



(a) 16×16×10 Image - Accuracy and Loss



(b) 20×20×16 Image - Accuracy and Loss

Figure 23 – 3D-CNN-SPP – Binary Cross-entropy using different scales

Image Scale	Accuracy	Loss
20×20×16	**96.41%**	**0.3922**
16×16×10	80.85%	0.3363

Table 8 – Accuracy & Loss – 3D-CNN-SPP x 3D-CNN

Although the 16×16×10 scale train and test converged better, the 20×20×16 scale provided remarkable results. Table 9 compares the Sensitivity, Precision, and FP_{avg} between the 3D-CNN and 3D-CNN-SPP models using 16 × 16 × 10 and 20 × 20 × 16 images. Figures 24 and 25 show the confusion matrix and AUROC curve, respectively, for 3D-CNN-SPP.

Figure 26 shows the feature mapping of positive (left) and negative (right) CMB detection.

Model/	**3D-CNN**		**3D-CNN-SPP**	
3D block size	**16 × 16 × 10**	**20 × 20 × 16**	**16 × 16 × 10**	**20 × 20 × 16**
Sensitivity	84.00%	95.00%	86.00%	97.00%
Precision	81.00%	94.00%	84.00%	95.00%
F1 score	81.00%	94.00%	84.00%	96.00%

Table 9 – 3D-CNN-SPP x 3D-CNN Metrics

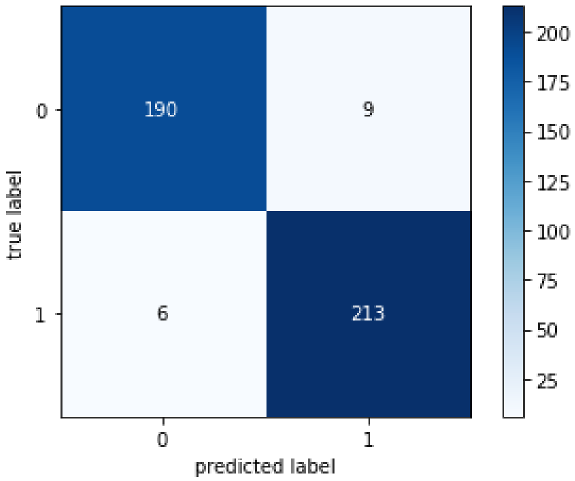


Figure 24 – 3D-CNN-SPP – Confusion Matrix

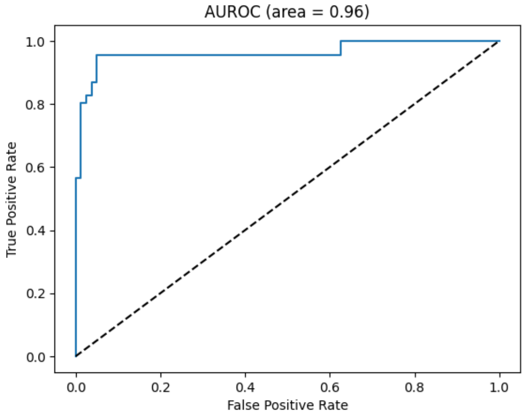


Figure 25 – 3D-CNN-SPP – AUROC

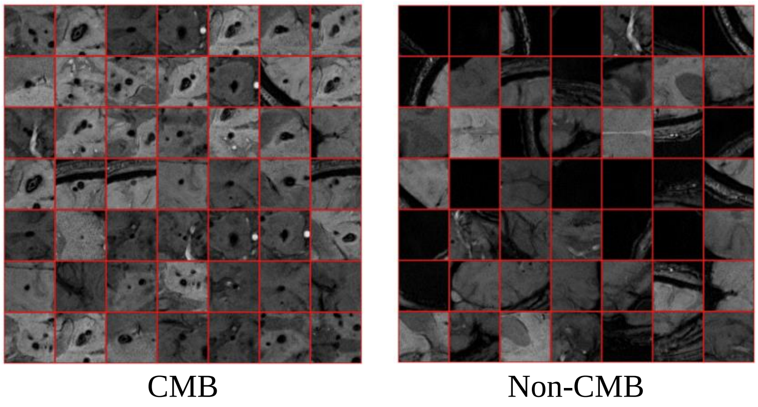


Figure 26 – 3D-CNN-SPP feature mapping

Finally, the study compares the results with the state-of-art approaches. The classification performance is given in Table 10. It must be noticed that the 3D-CNN-SPP improves precision compared to the 3D-CNN. The model's high sensitivity and precision indicate that it is making accurate and reliable predictions.

Method	Sensitivity	Precision	FP_{avg}
SVM	64.96%	5.13%	28.10
Random Forest	85.47%	17.24%	9.60
3D-CNN (Dou et al., 2016) (Original)	92.31%	42.69%	2.90
3D-CNN (Dou et al., 2016) (Reproduced)	94.45%	45.69%	2.32
3D-CNN-SPP	**96.94%**	**95.48%**	**9.00**

Table 10 – Performance comparison with state-of-the-art methods

Summary

This Chapter started by providing further details about the dataset used for the experiment. The Implementation and Hyperparameters Selection section briefly discusses architectural issues affecting the model and the need for regularization and learning parameters. It also addresses aspects of the optimizer’s parameters and how to prevent overfitting. The section ends by providing some rationale for small batch sizes use. The metrics section describes the formulas used to evaluate and compare the models. The 3D-CNN updated model section compares the original Dou et al. (2016) results with an updated version of the model to baseline the study. Finally, the Comparison with state-of-the-art methods section compares the 3D-CNN-SPP with the updated 3D-CNN model and provides a detailed comparison and analysis of the obtained results. It ends by comparing the 3D-CNN-SPP with other state-of-the-art methods for CMB classification.

Chapter 5

Conclusion

Conclusion

Cerebral microbleeds (CMBs) are small brain bleeding areas seen on magnetic resonance imaging (MRI) scans. They are believed to be caused by the breakdown of small blood vessels in the brain and are associated with an increased risk of stroke, dementia, and other neurological conditions.

As biomarkers, CMBs are essential because they can provide information about the underlying causes of these conditions and help identify individuals at risk for developing them. For example, CMBs are often found in individuals with hypertension, a leading stroke risk factor. By identifying individuals with hypertension who also have CMBs, healthcare providers can take steps to reduce their risk of stroke.

CMBs can also be used to monitor the progression of certain neurological conditions, such as Alzheimer's disease. For example, in individuals with Alzheimer's, CMBs may appear earlier, indicating that the disease progresses more quickly than in individuals without CMBs. In addition, CMBs can be used to evaluate the effectiveness of treatments for neurological conditions. For example, if an individual's CMBs remain stable or decrease after treatment, it may indicate that the treatment is working.

Overall, CMBs are important biomarkers because they provide valuable information about the underlying causes of neurological conditions, help identify individuals at risk for developing these conditions, and can be used to monitor the progression of these conditions and evaluate the effectiveness of treatments.

Considering the relevance of CMB diagnosis, this dissertation developed a 3D-CNN-SPP based on Dou et al. (2016) two-stage model. The suggested model improved the results obtained by the 3D-CNN method yielding an overall 96.94% sensitivity, 95.48% precision, and an f1 score of 96.20%. Another contribution of this work is providing the 3D-CNN-SPP with the capability to detect CMBs of different sizes and shapes. Additionally, this research conducted experiments to determine a viable choice of hyperparameters, addressed the small batch sizes issue for CNNs, and presented detailed results.

Suggestions for Future Work

Despite the satisfactory results obtained in this study, some problems remain unsolved. For example, selecting the parameters in the network is still time-consuming and dependent on the developer's experience. Therefore, other models with alternate architectures and hyper-parameters must be investigated further.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C. & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint airXiv:1603.04467.

Agarwal, A., Kanekar, S., & Mittal, V. (2016). Cerebral Microbleeds: Imaging Patterns, Interpretation, and Relevance. European Congress of Radiology 2016.

Akter, M., Hirai, T., Hiai, Y., Kitajima, M., Komi, M., Murakami, R. & Takahashi, M. (2007). Detection of hemorrhagic hypointense foci in the brain on susceptibility-weighted imaging: Clinical and phantom studies. *Academic Radiology*, *14*(9), 1011-1019.

Ateeq, T., Majeed, M. N., Anwar, S. M., Maqsood, M., Rehman, Z. U., Lee, J. W., ... & Mehmood, I. (2018). Ensemble-classifiers-assisted detection of cerebral microbleeds in brain MRI. *Computers & Electrical Engineering*, *69*, 768-781.

Basser, P. J., & Pierpaoli, C. (2011). Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *Journal of Magnetic Resonance, 213*(2), 560-570. doi:10.1016/j.jmr.2011.09.022

Charidimou, A., & Werring, D. J. (2012). Cerebral microbleeds and cognition in cerebrovascular disease: An update. *Journal of the Neurological Sciences*, *322*(1-2), 50-55.

Charidimou, A., Krishnan, A., Werring, D. J., & Jäger, H. R. (2013). Cerebral microbleeds: A guide to detection and clinical relevance in different disease settings. *Neuroradiology*, *55*(6), 655-674.

Charles, P.W.D. (2013), GitHub repository, Retrieved from https://github.com/keras-team/keras/tree/master/kerasgithub.com/charlespwd/project-title

Chen, H., Dou, Q., Wang, X., Qin, J., Cheng, J. C., & Heng, P. A. (2016). 3D fully convolutional networks for intervertebral disc localization and segmentation. In *International Conference on Medical Imaging and Augmented Reality* (pp. 375-382). Springer, Cham.

Ciresan, C.A., Giusti, L., Gambardella, M., & Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In *Medical image computing and computer-assisted intervention-MICCAI 2013* (pp. 411-418). Springer. New York.

Cohen-Adad, J., Descoteaux, M., Rossignol, S., Hoge, R., Deriche, R., & Benali, H. (2008). Detection of multiple pathways in the spinal cord white matter using q-ball imaging. *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. doi:10.1109/isbi.2008.4541091

Dou, Q., Chen, H., Yu, L., Shi, L., Wang, D., Mok, V. C., & Heng, P. A. (2015a). Automatic cerebral microbleeds detection from MR images via independent subspace analysis based on hierarchical features. In *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)* (pp. 7933-7936). IEEE.

Dou, Q., Chen, H., Yu, L., Shi, L., Mok, V. C., & Heng, P. A. (2015b). Automatic detection of cerebral microbleeds via deep-learning based 3d feature representation. In *2015 IEEE 12th international symposium on biomedical imaging (ISBI)* (pp. 764-767).

Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V.C., Shi, L. and Heng, P.A (2016). Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. *IEEE Transactions on Medical Imaging, 35*(5), 1182-1195.

Ghafaryasl, B., van der Lijn, F., Poels, M., Vrooman, H., Ikram, M. A., Niessen, W. J. & de Bruijne, M. (2012a). A computer-aided detection system for cerebral microbleeds in brain MRI. In *2012 9th IEEE international symposium on biomedical imaging (ISBI)* (pp. 138-141).

Ghafaryasl, B. (2012b). A computer-aided detection system for cerebral microbleeds in brain MRI. Master Dissertation, Kalmar University, Sweden

Goos, J. D., van der Flier, W. M., Knol, D. L., Pouwels, P. J., Scheltens, P., Barkhof, F., & Wattjes, M. P. (2011). Clinical relevance of improved microbleed detection by susceptibility-weighted magnetic resonance imaging. *Stroke*, *42*(7), 1894-1900.

Grauman, K., & Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (Vol. 2, pp. 1458-1465). IEEE.

Greenberg, S. M., Vernooij, M. W., Cordonnier, C., Viswanathan, A., Salman, R. A. S., Warach, S. & Microbleed Study Group. (2009). Cerebral microbleeds: A guide to detection and interpretation. *The Lancet Neurology*, *8*(2), 165-174.

Gregoire, S. M., Chaudhary, U. J., Brown, M. M., Yousry, T. A., Kallis, C., Jäger, H. R., & Werring, D. J. (2009). The Microbleed Anatomical Rating Scale (MARS): reliability of a tool to map brain microbleeds. *Neurology*, *73*(21), 1759-1766.

Griethuysen, J. J., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., ... & Aerts, H. J. (2017). Computational radionics system to decode the radiographic phenotype. Cancer research, 77(21), e104-e107.

Haacke, E. M., Mittal, S., Wu, Z., Neelavalli, J., & Cheng, Y. C. (2009). Susceptibility-weighted imaging: technical aspects and clinical applications, part 1. American Journal of Neuroradiology, 30(1), 19-30.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y. & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, *35*, 18-31.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904-1916.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2019). A survey of the recent architectures of deep convolutional neural networks. *arXiv preprint arXiv:1901.06032*.

Kolarik, M., Burget, R., & Riha, K. (2020). Comparing normalization methods for limited batch size segmentation neural networks. *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. https://doi.org/10.1109/tsp49548.2020.9163397

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Kuijf, H. J., de Bresser, J., Biessels, G. J., Viergever, M. A., & Vincken, K. L. (2011). Detecting cerebral microbleeds in 7.0 T MR images using the radial symmetry transform. In *2011 IEEE international symposium on biomedical imaging: from nano to macro* (pp. 758-761). IEEE.

Kuijf, H. J., Brundel, M., de Bresser, J., van Veluw, S. J., Heringa, S. M., Viergever, M. A., ... & Vincken, K. L. (2013). Semi-automated detection of cerebral microbleeds on 3.0 T MR images. *PLoS One*, *8*(6).

Lazebnik, S., Schmid, C., & Ponce, J. (2006, June). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 2, pp. 2169-2178). IEEE.

Li, J., Si, Y., Lang, L., Liu, L., & Xu, T. (2018). A spatial pyramid pooling-based deep convolutional neural network for the classification of electrocardiogram beats. *Applied Sciences*, *8*(9), 1590.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.

Lo, S. C. B., Chan, H. P., Lin, J. S., Li, H., Freedman, M. T., & Mun, S. K. (1995). Artificial convolution neural network for medical image pattern recognition. *Neural Networks*, *8*(7-8), 1201-1214.

Loy, G., & Zelinsky, A. (2003). Fast radial symmetry for detecting points of interest. *IEEE Transactions on pattern analysis and machine intelligence*, *25*(8), 959-973.

Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *ArXiv, abs/1804.07612*.

Menze, B. H. et al. (2015). "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," in IEEE Transactions on Medical Imaging, vol. 34, no. 10, pp. 1993-2024, Oct. 2015.

Newman, T. S., Tang, N., Bacharach, S., & Choyke, P. (1996). A volumetric segmentation technique for diagnosis and surgical planning in lower torso CT images. In Proceedings of 13th International Conference on Pattern Recognition (Vol. 3, pp. 553-557). IEEE.

Ouyang, X., Gu, K., & Zhou, P. (2018). Spatial pyramid pooling mechanism in 3D convolutional network for sentence-level classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(11), 2167-2179.

Qu, T., Zhang, Q., & Sun, S. (2017). Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks. *Multimedia Tools and Applications*, *76*(20), 21651-21663.

Rao, M., Tang, P., & Zhang, Z. (2020). A Developed Siamese CNN with 3D Adaptive Spatial-Spectral Pyramid Pooling for Hyperspectral Image Classification. *Remote Sensing, 12*(12), 1964. doi:10.3390/rs12121964

Roth, H. R., Lu, L., Farag, A., Shin, H. C., Liu, J., Turkbey, E. B., & Summers, R. M. (2015). Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 556-564). Springer, New York.

Sabiri, B., El Asri, B., & Rhanoui, M. (2022). Mechanism of overfitting avoidance techniques for training deep neural networks. *Proceedings of the 24th International Conference on Enterprise Information Systems*. https://doi.org/10.5220/0011114900003179

Santos, C. F., & Papa, J. P. (2022). Avoiding overfitting: A survey on regularization methods for Convolutional Neural Networks. *ACM Computing Surveys*, *54*(10s), 1–25. https://doi.org/10.1145/3510413

Seghier, M., Kolanko, M. A, Leff, R., Jäger, H. R., Simone, G., Werring, D. (2011). Microbleed detection using automated segmentation (MIDAS): A new method applicable to standard clinical MR images. *PloS ONE.*

Shanmugamani, R. (2018). *Deep Learning for computer vision: expert techniques to train advanced neural networks using TensorFlow and Keras*. Birmingham, UK: Packt Publishing.

Tariq, H., & Burney, A. (2014). Brain MRI literature review for interdisciplinary studies. Journal of Biomedical Graphics and Computing, 4(4), 41-53.

Tao, Y., & Cloutie, R. S. (2018). Voxelwise Detection of Cerebral Microbleed in CADASIL Patients by Genetic Algorithm and Back Propagation Neural Network. In *2018 3rd International Conference on Communications, Information Management and Network Security (CIMNS 2018).* Atlantis Press.

Yang, J., Jiang, Y. G., Hauptmann, A. G., & Ngo, C. W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on multimedia information retrieval* (pp. 197-206).

Zhang, Y., Zhang, E., & Chen, W. (2016). Deep neural network for halftone image classification based on sparse auto-encoder. *Engineering Applications of Artificial Intelligence*, *50,* 245-255.

Zhang, L., Zhong, Y., Zhao, B., & Han, X (2017). Scene classification based on a hierarchical convolutional sparse auto-encoder for high spatial resolution imagery. *International Journal of Remote Sensing*, *38*(2), 514-536.

Zhang, Y. D., Zhang, Y., Hou, X. X., Chen, H., & Wang, S. H. (2018). Seven-layer deep neural network based on sparse autoencoder for voxelwise detection of cerebral microbleeds. *Multimedia Tools and Applications*, *77*(9), 10521-10538.