

Automatic Identification of Extremely Tiny Brain Hemorrhages in Susceptibility Weighted Images using Convolutional Neural Network

Sahereh Obeidavi
Chemnitz University of Technology,
Reichenhainer Straße 70, Chemnitz, Germany

ABSTRACT

Ischemic stroke is an acute cerebrovascular disease that causes long-term disability and even death. Acute lesions that occur in most stroke patients can be eliminated with careful diagnosis and treatment. The presence of acute lesions in a majority of stroke cases necessitates precise diagnosis and treatment for elimination. Despite the sensitivity of MRI imaging to these lesions, accurately gauging their location and volume manually poses challenges for physicians. The manual examination of numerous MRI-generated cross-sections is time-consuming and susceptible to human error. Consequently, the consensus among medical practitioners is that automated segmentation procedures for ischemic stroke lesions can significantly expedite the commencement of treatment. Various methods have been developed to attain this objective, with deep neural networks emerging as notably effective, producing outcomes that are both superior and more precise. Within the realm of deep learning algorithms, the U-Net algorithm has gained popularity in recent years for its accurate response, high precision, rapid processing and learning capabilities, and its independence from large datasets for learning. The U-Net algorithm has become a favored choice for identifying and segmenting image components in the processing of medical images. The proposed segmentation framework comprises two distinct networks: the U-Net convolutional neural network serves as the primary structure of the model, while the Inception convolutional neural network is integrated into each layer of the U-Net network. Incorporating the Inception network within the U-Net network has notably enhanced segmentation accuracy. This report focuses on elucidating the algorithm's intricacies, encompassing its architectures, pre-processing techniques, data pre-preparation, and post-processing methods. The structural aspects of the algorithm, particularly its convolutional network, are explored in depth. Additionally, the optimal configuration for the algorithm's parameters and super parameters is investigated to enhance and achieve peak accuracy in the segmentation of stroke-related images.

General Terms

Medical Imaging, Ischemic Stroke, MRI Segmentation, Brain Hemorrhages, Deep Learning Algorithms, Convolutional Neural Networks (CNN), U-Net Algorithm, Inception Network, Image Processing, Automated Segmentation, Susceptibility Weighted Images, Diagnostic Imaging

Keywords

Machine Learning, Convolutional Neural Network, Brain Hemorrhages, Stroke, Classification.

1. INTRODUCTION

Very small accumulations of blood or Cerebral microbleeds (CMB) in the brain are more commonly found in patients with stroke, dementia, and cardiovascular disease [1]. The diagnosis of CMBs can contribute to predicting stroke. Furthermore, recent clinical studies have underscored that Cerebral Microbleeds (CMBs) can contribute to cognitive impairment, including conditions like dementia [2]. Magnetic Resonance Imaging (MRI) is employed for CMB detection, leveraging its advantages over alternative imaging modalities. Among the various methods of magnetic resonance imaging, two techniques, T2-star and 2SWI, are utilized. [3] showed that Susceptibility weighted imaging (SWI) increases the number of CMBs detected.

Image processing for automatic detection of CMBs faces challenges: a) different sizes of CMBs between 2 and 10 mm, b) different locations of CMBs [1], and c) there are many pseudo-CMBs, that makes the diagnosis difficult. In current clinical practice, CMBs are labeled manually. [4] Manual labeling proves to be difficult, time-consuming, and error-prone. Consequently, numerous automated and semi-automated algorithms have been developed. Existing algorithms exhibit a substantial number of positive false positives (FP), diminishing their value and necessitating ongoing research in this domain. The following section reviews some examples of prior works.

Reference [5], provides a semi-automated method for detecting CMBs. In this study, a threshold algorithm is initially employed to distinguish CMBs from pseudo-CMBs. Subsequently, a support vector classifier (SVM) (4) is utilized, and finally, manual post-processing is conducted. Reference [6] employed a random forest classifier and achieved a sensitivity of 92.04%. The extracted properties were then classified using the 5ISA network and support vector machine classifier with 89.44% sensitivity. In reference [8], a new structure of convolutional neural network (CNN) with a 6RBAP layer was presented, achieving an accuracy of 97.18%. In reference [9], optimal convolutional neural network parameters led to a sensitivity of 99.74%, while in reference [10], a method based on ResNet-50 was proposed, reporting a sensitivity of 95.71%.

The paper endeavors to enhance the evaluation parameters for CMB detection through various experiments on adjustable convolutional neural network parameters. Ultimately, a network with three convolution layers, two pulling layers, and one fully connected layer is proposed.

The present manuscript is structured into four sections: The second section introduces the proposed algorithm. The third section covers experiments and the evaluation of the proposed

method, while the fourth section provides a summary of the work.

2. PROPOSED CNN NETWORK STRUCTURE

To address the challenge of detecting Cerebral Microbleeds (CMB) using SWI images, a CNN network has been introduced

as a proposed solution. The algorithm outlined encompasses a pre-processing step and the integration of a convolutional neural network. In efforts to enhance the CNN network's performance, a series of experiments were conducted to assess factors influencing its effectiveness. The parameters for evaluating the network structure were then selected based on the obtained results. The illustrated CNN network structure is presented in "Figure 1".

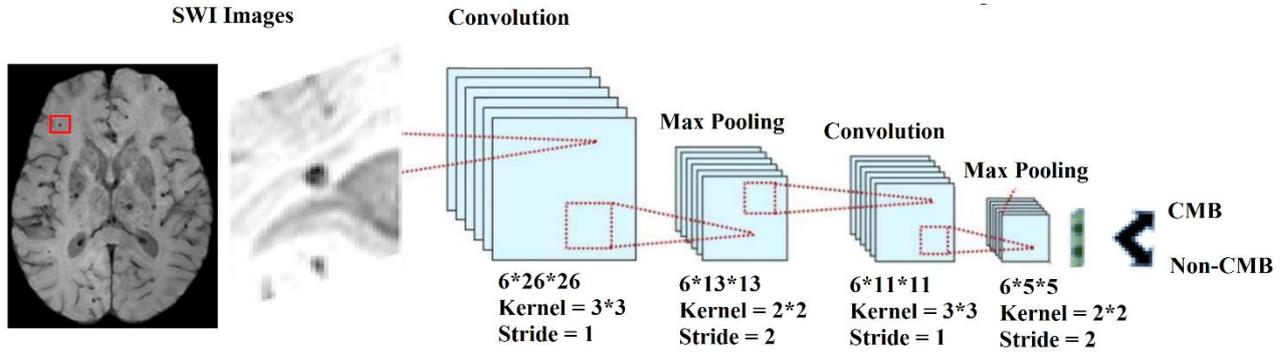


Fig 1: The proposed CNN network structure

Next, convolution and preprocessing neural networks were described as prerequisites.

2.1 Convolution layer

The convolutional layer serves as the primary network layer, housing a collection of adaptable filters. Each filter is spatially compact but extends across the depth of the input mass. In simpler terms, a filter is a three-dimensional structure defined by a specific number. However, it's important to note that a higher number of filters increases the computational load.

The activator function applied to the convolution layer's output can be either linear or nonlinear. Nonlinear functions are employed to distinguish data that lacks linear separability. In this study, Rectified Linear Unit (ReLU), Leaky ReLU, and Parametric ReLU (PReLU) functions were investigated as activation functions, defined by equations (1), (2), and (3), respectively.

$$y = \max(0, x) \quad (1)$$

$$y = \begin{cases} 0.01x & x < 0 \\ x & x > 0 \end{cases} \quad (2)$$

$$y = \begin{cases} ax & x < 0 \\ x & x > 0 \end{cases} \quad (3)$$

2.2 Max pooling layer

The max pooling layer is usually placed after the convolution layer, and by maximizing 8 or averaging 9, the number of pixels in the width and height of the convolved feature map will be reduced, followed by a decrease in the number of parameters and the complexity of the calculations. The pooling operation is executed independently at each depth.

2.3 Fully connected layer

True to its name, every neuron in this layer establishes connections with neurons in the preceding layer. The fully connected layer amalgamates all features to categorize the input image.

2.4 Preprocessing

In the preprocessing step, as described by relation (4), normalization is applied to the input data represented by X_0 . The normalization formula is given by.

$$X = \frac{X_0}{\max(X_0)} \quad (4)$$

This normalization process ensures that the maximum pixel size in the input data is limited to one. The primary objective of normalization is to bring uniformity to the distribution of input pixels. This uniformity is crucial for enhancing the training process of the neural network, leading to faster convergence.

The convolutional neural network (CNN) proposed for this study comprises a total of 6 layers. The architecture of the network is summarized in "Table 1," detailing the layers, filter sizes, filter numbers, and output dimensions at each stage. These layers collectively form the architecture of the proposed CNN. Each convolutional layer is followed by a LeakyReLU activation function, and specific pooling layers contribute to dimensionality reduction at strategic points in the network. This architecture is designed to capture hierarchical features in the input data, leading to effective representation learning.

Table 1. Proposed network layers

No.	Layers	Filter size	Filter number	Output dimensions
1	Pre-processed input patch	-	-	(61,61,1)
2	Convolution + LeakyReLU	11x11	32	(61,61,32)
3	Maximum pooling	-	-	(30,30,32)
4	Convolution + LeakyReLU	3x3	32	(30,30,32)

5	Maximum pooling	-	-	(15,15,32)
6	Convolution + LeakyReLU	3×3	32	(15,15,32)

3. TESTS AND EVALUATIONS

In this part of the research, the database for network training, evaluation criteria, and results are introduced.

Implementations in the Python programming language are done on Google Colab.

For validation in this research, the 10-folder cross-validation method has been used. In this method, the database is randomly divided into 10 subsets. In each subset, some data is stored as validation data for model testing, and the rest of the data is used as training data. The results are then averaged to produce a single estimate.

3.1 Data Preparation

In this study, 20 volumes of SWI-CMB database images that are available to the public [11] have been used. Most images are $150 \times 512 \times 512$ in size. By separating and saving the slides from the image size, 2982 images were created, which is 63 images from this CMB collection. 10SNP [12] with a $61 \times 61 \times 61$ slider was used to create input patches and tags. The reason for choosing the $61 \times 61 \times 61$ size for the patch is that this size preserves useful information for detection [9]. The central pixels of the patch, which contained CMB, was labeled one and non-CMB labeled zero. "Figure 2" and "Figure 3" show examples of patches labeled one and zero, respectively.

Because there were several CMBs in some of the CMB images, the number of CMB patches was reduced to 74 patches after the separation of these CMBs, and to 1776 patches by adding plugin data from symmetry, rotation, and displacement.

Non-CMB images yielded 1500 non-CMB patches. In total, 3276 patches were employed for training and evaluating the network, with 90% allocated for training and the remaining 10% for evaluation. It is crucial to emphasize that plugin data for the test set has been excluded from this analysis.

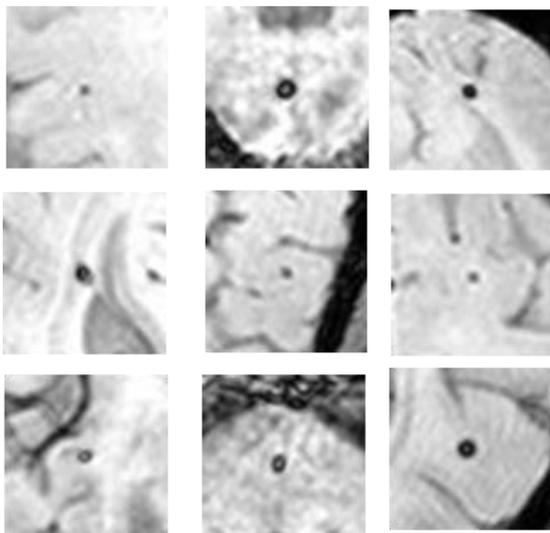


Fig 2: Sample CMB patches

Empirical findings have demonstrated superior outcomes with the Leaky-ReLU activator function and the max-pooling layer. Consequently, these elements were incorporated into the proposed network. To finalize the classification of input images into CMB or non-CMB categories, a fully connected layer was positioned at the end, guided by the training algorithm.

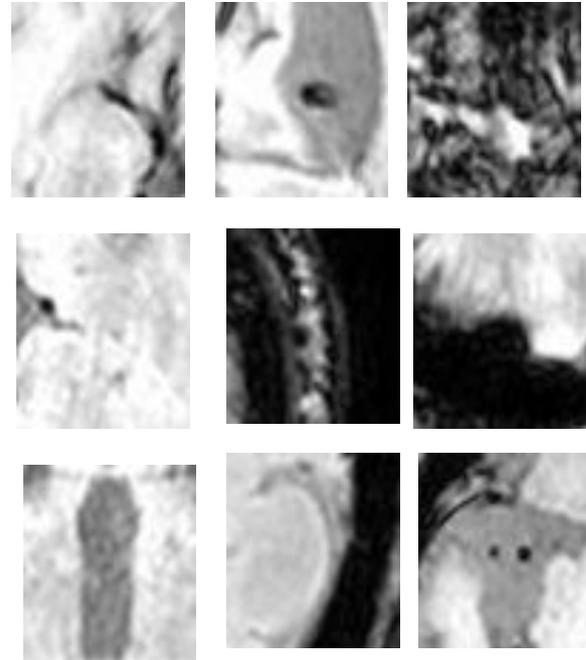


Fig 3: Sample non-CMB patches

3.2 Evaluation parameters

To evaluate the classification results, three indicators were used: sensitivity, specificity and accuracy. The mentioned indicators are commonly used metrics to assess the performance of a classification model. In the context of this study, positive and negative refer to the classes "CMB" (cerebral microbleeds) and "non-CMB" (no cerebral microbleeds), respectively.

Sensitivity (True Positive Rate) is a measure of how well the model identifies instances of the positive class (CMB). It is calculated using the formula (5). Sensitivity, also known as the True Positive Rate or Recall, tells the proportion of actual positive instances correctly identified by the model.

Specificity (True Negative Rate) measures the model's ability to correctly identify instances of the negative class (non-CMB). It could be calculated as formula (6). Specificity indicates the proportion of actual negative instances correctly identified by the model.

Accuracy provides an overall assessment of the model's correctness by considering both true positives and true negatives, as well as false positives and false negatives. The formula for accuracy is equation (7). Accuracy gives the percentage of correctly classified instances among all instances.

These metrics are valuable for assessing the performance of classification model, providing insights into its strengths and weaknesses in differentiating between CMB and non-CMB cases.

$$\text{sensitivity} = \frac{TP}{(TP + FN)} \quad (5)$$

$$\text{specificity} = \frac{TN}{(TN + FP)} \quad (6)$$

$$\text{accuracy} = \frac{(TP + TN)}{(TP + TN + FN + FP)} \quad (7)$$

3.3 Experiments

A series of experiments were meticulously conducted to attain the desired model structure and comprehensively assess the effectiveness of the proposed algorithm. The optimization process employed the Adam optimizer, known for its efficiency in terms of low memory usage and rapid convergence. This choice of optimizer was crucial to enhance the overall training efficiency of the algorithm.

The cost function, a pivotal aspect of the research methodology, was formulated as a reciprocal entropy function. This particular choice of cost function holds significance in guiding the model towards optimal convergence, aligning with the goals of the study.

The Convolutional Neural Network (CNN) underwent initialization with random weights, a crucial step in the training process. Notably, the adjustment of initial weights extended to 13, contributing to the model's capacity to learn and adapt to the intricacies of the dataset. Furthermore, a batch size of 128 was employed during the training process, facilitating efficient updates to the model parameters and enhancing the convergence speed.

In the fully connected layer of the CNN, the softmax function was employed. The softmax function, a key component in multiclass classification tasks, facilitated the conversion of raw model outputs into probability distributions over multiple classes. This utilization of the softmax function added a layer of interpretability to the model's predictions.

In summary, the experimental setup encompassed the use of the Adam optimizer, a carefully selected cost function, strategic initialization of CNN weights, and the incorporation of the softmax function in the fully connected layer. These choices were made with the intent of optimizing the algorithm's performance, achieving the desired model structure, and ensuring a thorough evaluation of its capabilities.

- **Experiment I: Set the number of IPAC algorithms**

In this experiment, the algorithm was executed 10 times per number of different IPACs and the average evaluation criteria per 10 times performed are shown in "Figure 4". As can be seen from "Figure 4", the number of ipak equals 11 will be a good choice.

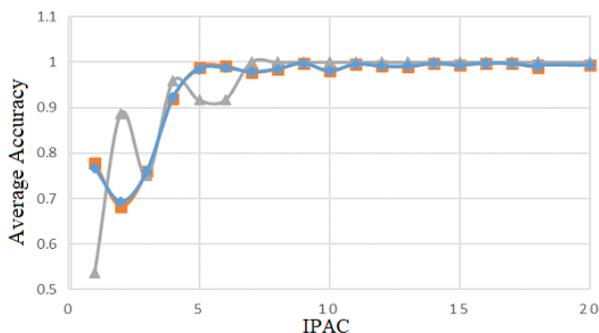


Fig 4: Set the number of IPAC algorithms

- **Experiment II: Adjust the number of layers**

Research shows that the number of layers can affect CNN performance, especially convolutional layers used to extract features. The average results obtained from 10 executions for different layer arrangements are given in "Table 2". As can be seen from "Table 2", the arrangement of the layers in the form of Conv / Pl / Conv / Pl / Conv, respectively, will be a good choice.

Table 2. The effect of layer layout on CNN network performance.

Layers	Average Accuracy (%)	Average Feature (%)	Average Sensitivity (%)
Conv	96/32	97/46	75
Conv/Pl	97/34	98/4	77/5
Conv/Pl/Conv	97/34	98/2	81/25
Conv/Pl/Conv/Pl	98/10	98/1	95
Conv/Pl/Conv/Pl/Conv	98/3	98/2	96/25
Conv/Pl/Conv/Pl/Conv/Pl	98/35	98/2	95

- **Experiment III: Activation and pooling function**

Diverse combinations of activation and pooling functions result in varied network behaviors, necessitating the testing of various combinations. The algorithm underwent 10 iterations for each pairing of activation and pooling functions, with the mean values of the evaluation criteria presented in "Table 3." As per the findings in "Table 3", the effectiveness of the average-based pooling layer is inferior to that of the max-based pooling layer.

Table 3. The effect of different combinations of activation and pooling functions on CNN network performance.

Activator Function	Pooling based on	Average accuracy (%)	Average feature (%)	Average sensitivity (%)
ReLU	Max.	98/67	98/8	96/25
PReLU	Max.	97/4	97/6	93/75
Leaky ReLU	Max.	99/24	99/26	98/75
ReLU	Average	97/9	98/2	92/5
PReLU	Average	96/01	96/53	86/25
Leaky ReLU	Average	97/84	97/8	98/75

- **Experiment IV: Size and number of filters in the convolution layer**

In the first convolution layer, to achieve the best detection performance of CMBs, different numbers and sizes of the corresponding filter in this layer were examined, the algorithm was run 10 times per mode, and the mean values of the evaluation parameters in “Table 4” and “Table 5” is brought. Thus, as shown in these Tables, the filter size was set to 5 and the number of filters to 32.

Table 4. The effect of first convolution layer filter size on CNN network performance.

Filter size	Average accuracy (%)	Average feature (%)	Average sensitivity (%)
3	98/98	99	98/75
5	99/43	99/46	99
7	99/05	99	99
9	99/05	96/06	98/75
11	98/98	96/06	97/5

Table 5. The effect of the number of first convolution layer filters with dimensions of 5*5 on the efficiency of the CNN network.

Num. of Filters	Average accuracy (%)	Average feature (%)	Average sensitivity (%)
8	98/48	98/73	93/75
16	99/17	99/13	98/91
32	99/22	99/2	99/9
64	99/24	99/2	99/8
128	98/92	98/86	99/93

- **Experiment V: The proposed network performance**

In order to improve the classification performance of CMBs, experiments were performed to test almost all the factors that could affect CNN performance. Finally, a convolutional neural network consisting of three convolutional layers, two pulling layers, and a fully connected layer achieved better performance. To avoid randomness, the evaluation results were performed 10 times. The mean and deviation from the mean are shown in “Figure 5”.

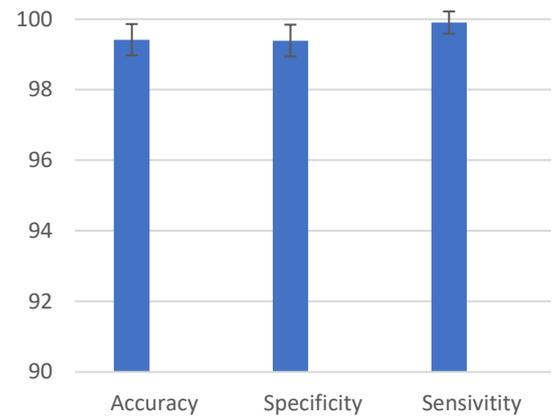


Fig 5: Evaluation results in 10 times the implementation of the proposed network.

The efficiency of the proposed network with other methods is shown in “Table 6”. As can be seen, the proposed method has better results than other methods.

Table 6. Comparison of the proposed method with previous articles.

-	[8]	[9]	[10]	Proposed method
Year	2017	2018	2019	2022
Images format	SWI	SWI	SWI	SWI
Accuracy (%)	97/18	98/32	97/46	99/41
Feature (%)	97/18	96/89	99/21	99/39
Sensitivity (%)	96/94	99/74	95/71	99/9

4. CONCLUSION

In conclusion, this paper introduces an automated approach for detecting cerebral hemorrhages, leveraging the robust capabilities of convolutional neural networks. The necessity for automation arises from the inherent challenges associated with manual processing, including its difficulty, time-intensiveness, and susceptibility to errors. The pivotal step of image feature extraction is addressed through the application of a 6-layer convolutional neural network with compact dimensions, striking a balance between reducing network parameters and maintaining accuracy for classification.

The experiments conducted on the SWI image database affirm the efficacy of the proposed method, showcasing superior results compared to reference methods. The demonstrated success paves the way for broader applications in medical image analysis, particularly in the early and accurate identification of cerebral hemorrhages.

Looking ahead, the future scope of this research encompasses further refinement and extension of the proposed algorithm. Continuous exploration and optimization of convolutional neural network architectures, activation functions, and pre-processing techniques can contribute to even more precise and efficient hemorrhage detection. Additionally, the integration of emerging technologies such as explainable artificial intelligence and transfer learning could enhance the interpretability and generalizability of the model.

Furthermore, the adaptability of the algorithm to diverse medical imaging datasets beyond SWI, along with its potential integration into real-time clinical workflows, remains a promising avenue for future investigation. Collaborations with healthcare professionals and experts in the field can provide valuable insights for refining the algorithm's applicability and ensuring its seamless integration into clinical practice.

In conclusion, while this paper addresses a critical aspect of automated cerebral hemorrhage detection, the journey continues towards advancing and fine-tuning the proposed method, fostering a positive impact on the realm of medical image analysis and contributing to improved patient outcomes.

5. REFERENCES

- [1] Wu, Y. and Chen, T. (2016). An up-to-date review on cerebral microbleeds. *Journal of Stroke and Cerebrovascular Diseases*, 25(6):1301–1306.
- [2] Charidimou, A., Krishnan, A., Werring, D. J., and Rolf J'ager, H. (2013). Cerebral microbleeds: a guide to detection and clinical relevance in different disease settings. *Neuroradiology*, 55:655–674.
- [3] Shams, S., Martola, J., Cavallin, L., Granberg, T., Shams, M., Aspelin, P., Wahlund, L., and Kristoffersen-Wiberg, M. (2015). Swi or t2*: which mri sequence to use in the detection of cerebral microbleeds? The karolinska imaging dementia study. *American Journal of Neuroradiology*, 36(6):1089–1095.
- [4] De Bresser, J., Brundel, M., Conijn, M., Van Dillen, J., Geerlings, M., Viergever, M., Luijten, P., and Biessels, G. (2013). Visual cerebral microbleed detection on 7t mr imaging: reliability and effects of image processing. *American Journal of Neuroradiology*, 34(6):E61–E64.
- [5] Barnes, S. R., Haacke, E. M., Ayaz, M., Boikov, A. S., Kirsch, W., and Kido, D. (2011). Semiautomated detection of cerebral microbleeds in magnetic resonance images. *Magnetic resonance imaging*, 29(6):844–852.
- [6] Fazlollahi, A., Meriaudeau, F., Villemagne, V. L., Rowe, C. C., Yates, P., Salvado, O., and Bourgeat, P. (2014). Efficient machine learning framework for computer-aided detection of cerebral microbleeds using the radon transform. In *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*, pages 113–116. IEEE.
- [7] Dou, Q., Chen, H., Yu, L., Shi, L., Wang, D., Mok, V.C., and Heng, P. A. (2015). Automatic cerebral microbleeds detection from mr images via independent subspace analysis based hierarchical features. In *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pages 7933–7936. IEEE.
- [8] Wang, S., Jiang, Y., Hou, X., Cheng, H., and Du, S. (2017). Cerebral micro-bleed detection based on the convolution neural network with rank based average pooling. *IEEE Access*, 5:16576–16583.
- [9] Hong, J., Wang, S.-H., Cheng, H., and Liu, J. (2020). Classification of cerebral microbleeds based on fully-optimized convolutional neural network. *Multimedia Tools and Applications*, 79:15151–15169.
- [10] Hong, J., Cheng, H., Zhang, Y.-D., and Liu, J. (2019). Detecting cerebral microbleeds with transfer learning. *Machine Vision and Applications*, 30:1123–1133.
- [11] Q 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.
- [12] Zhang, Y.-D., Hou, X.-X., Chen, Y., Chen, H., Yang, M., Yang, J., and Wang, S.-H. (2018). Voxelwise detection of cerebral microbleed in cadasil patients by leaky rectified linear unit and early stopping. *Multimedia Tools and Applications*, 77:21825–21845.