2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)
Manipal University Jaipur, Sep 28-29, 2019

Brain Stroke Detection Using Convolutional Neural Network and Deep Learning Models

Bhagyashree Rajendra Gaidhani[1]
Post Graduate Scholor
Sandip university, Nashik
gaidhanibhagyashree@gmail.com

Dr. R.Rajamenakshi[2]
Joint Director
CDAC, Pune
menakshi@cdac.in

Samadhan Sonavane[3]
Assistant Professor
Sandip University, Nashik
samadhan.sonavane@sandipuniversity.edu.in

***Abstract*- For the last few decades, machine learning is used to analyze medical dataset. Recently, deep learning technology gaining success in many domain including computer vision, image recognition, natural language processing and especially in medical field of radiology. This research attempts to diagnose brain stroke from MRI using CNN and deep learning models. The proposed methodology is to classify brain stroke MRI images into normal and abnormal images and delineate abnormal regions using semantic segmentation [4]. In particular, two types of convolutional neural network that are LeNet [2] and SegNet are used. For classification, we passed pre-processed stroke MRI for training, trained all layers of LeNet and classify normal and abnormal patient. Then this abnormal patient data stored into two dimensional array and passed this two dimensional array to SegNet which is auto encoder decoder [3] model for segmentation, trained all layers of SegNet except fully connection layer. The experimental result show that classification model achieve accuracy between 96-97% and segmentation model achieve accuracy between 85-87%.Through experimental results, we found that deep learning models not only used in non-medical images but also give accurate result on medical image diagnosis, especially in brain stroke detection.**

Keywords-brain stroke, deep learning, convolutional neural network, semantic segmentation.

I. INTRODUCTION

The term "stroke" encompasses both ischemic and haemorrhage disturbances of the cerebral circulation producing central neurological deficits of acute or sub-acute onset [1]. Ischemia accounts for 80 to 85% of stroke haemorrhage for 15 to 20%.

The effects of a stroke are unique to each individual, and recovering from a stroke is different for each person. To diagnose stroke patient, radiologist use CT, MRI scan but, sometimes radiologist find difficult to identify abnormality in images, computer aided diagnosis (CAD) plays important role in medical image analysis which helps radiologist to evaluate and analyze abnormality in short time. In last few decades, machine learning algorithm have been extensively used in computer-aided diagnosis (CAD) for the detection of critical of diseases such as lung cancer, breast cancer, brain tumor etc.

Recently, deep learning [7][9] have shown revolutionary performance in variety of task, especially related to images where networks learn large number of dataset, extract features from images automatically and use interconnected units to solve complex task. Since the medical field of radiology mostly relies on extracting patterns from images, it is natural application of deep learning [10] and research in this area has rapidly grown in recent years.

II. RELATED WORK

The increasing growth of deep learning, computer vision techniques divided into traditional methods and deep learning methods. This section describe related work of brain stroke and how deep learning methods are better than conventional methods.

In 2013, HemaRajini and Bhavani [5] published a paper on automatic detection of ischemic stroke, paper presented an approach that separates normal tissue from abnormal tissue using segmentation. The method in this paper is divided into four stages, pre-processing, tracking brain midline, feature extraction and classification. The SVM, KNN are used for classification. But it requires large memory and result is not accurate.

In 2011, Fuk-hay Tang et al [6] published a paper on detecting early ischemic stroke and brain lesions. Used Otsu algorithm for binarizing images and features passed through Circular Adaptive Region of Interest to find out location of lesion in image, but the number of samples used are too small. Sometimes it is not giving effective result on unseen data.

To overcome above disadvantages. Recently, Deep learning algorithms have shown revolutionary performance in different task, especially for analyzing medical images. Convolutional neural network is a deep learning algorithm mainly used in computer vision task including radiology and it is designed to learn spatial features through backpropogation using multiple building blocks to obtain outstanding effect in image recognition.

III. OBJECTIVE OF THIS RESEARCH

The objective of this study to develop fast and reliable method which automatically diagnose brain stroke and delineate abnormal regions accurately. To design automated system, LeNet (convolutional neural network model) will be used to determine

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.

existence of brain stroke. Then, used SegNet for semantic segmentation which delineate abnormal regions.

The rest of this paper is organized as follows, section II presents proposed methodology in that classification and segmentation models, section III presents experiment result and analysis followed by conclusion and references.

IV. PROPOSED METHODOLOGY

This research proposed two models, one for classification and one for segmentation. LeNet (convolutional neural network) model used for classification, which classified result into normal and abnormal tissue. Then passed all abnormal images and their labels for Segmentation. For segmentation SegNet (Auto encoder decoder) model is used. Segmentation delineate abnormal region of interest from MRI. This section presents step by step process of classification and segmentation models. The proposed methodology shown in figure 1.

A. Data Acquisition

The brain stroke data were obtained from ATLAS (Anatomical Tracings of Lesions after Stroke), website mainly for neuroimaging studies. By agreeing terms of condition it was possible to download brain stroke dataset. The dataset contained stroke anatomical brain images and manually lesion segmentation. The data was in Nifti format and contained around 229 T1-weighted MRI scans with manual segmentation and metadata in .csv format and normal patient dataset contained around 400 MRI scans.

B. Data pre-processing

3D image series was resized stacked as 2D- array. Image pre-processing is significant in medical image analysis. To preserve image quality image denoising and enhancement technique are used. Median filter was used to remove noise generated. We replaced each pixel value with median of its neighbors and pixel which are very different from neighboring pixel are eliminated. Then perform normalization on data by rescaling pixel range in between 0 and 1, took minimum and maximum value from images, then transform N-dimensional image into new intensity value in the range(new_min, new_max)I_{New}= {new_min ...new_max}. The normalization is calculated as given in equation (1).

$$I(new) = \frac{I - I(min)}{I\,(max) - I\,(min)} \tag{1}$$

C. Phase I: Classification using LeNet

i. Model building

For classification we used LeNet which is convolutional neural network architecture[8].It is small and can run on CPU (if we don't have GPU support) and it is 7 –layer architecture consists of 3 convolution layer, 2 max-pooling layer, 1 fully connected layer and 1 output layer. The dataset used for classification is normal and abnormal images. The network consist of 3 convolutional layer, 2 max-pooling layer, 1 fully connected layer and 1 output layer. Convolutional layer extract features from image using filters. The filters are convolved over input volume, learned relationship between pixels and preserve it into feature map.

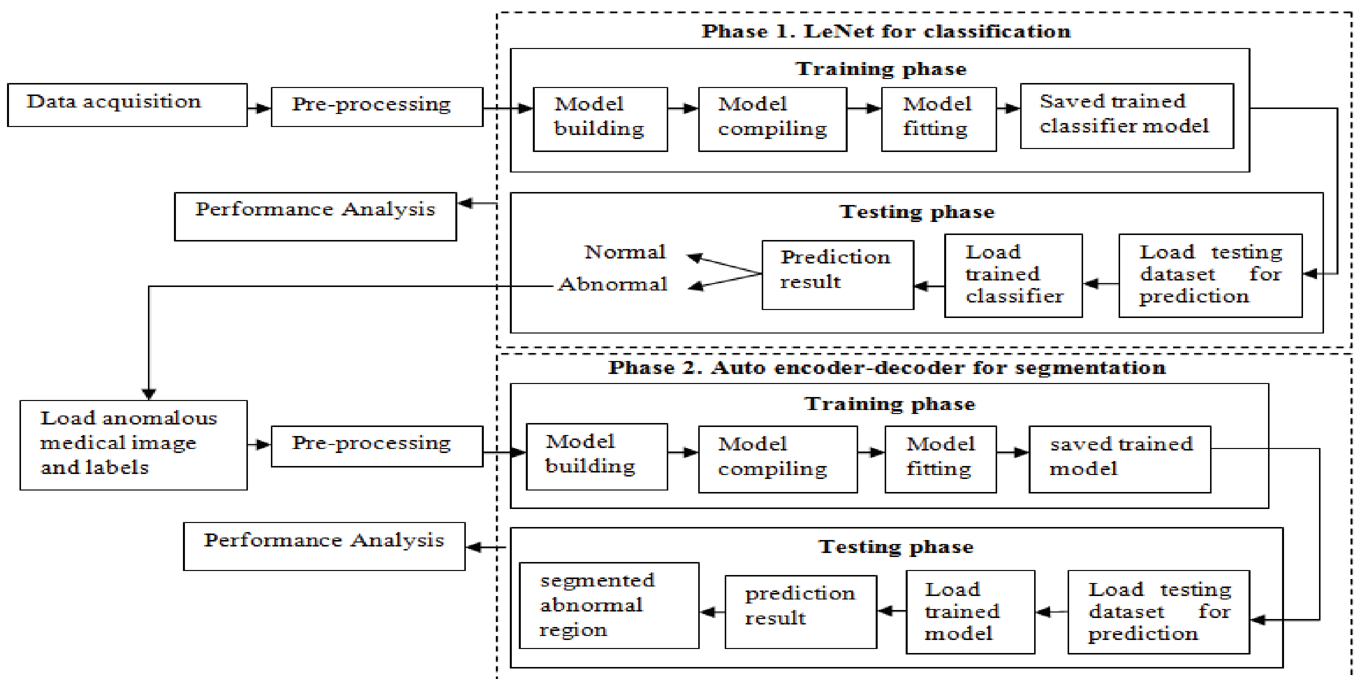


Fig. 1. Proposed Methodology

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.

Output of this convolutional layer is passed to the ReLu activation followed by dropout layer for avoiding overfitting. The Relu activation function convert negative value neuron into zero and only activate positive value neuron. Relu activation function is calculated based on equation (2).

R(x) =max (0, x) where, it gives output when x is positive otherwise return zero.

$$f(x) = \begin{cases} 0 \text{ for, } x < 0 \\ x \text{ for, } x \geq 0 \end{cases} \quad (2)$$

Only neuron has positive value are passed to the max pooling layer. Pooling operation reduce dimensionality of images by passing window over an image according to stride and at each step, value within the window is extracted into output matrix, data was reshape by setting padding parameter "same" and passed to fully connected layer. The fully connected layer flatten output matrix into vector and passed this vector to the softmax function as given in equation (3). This function calculates probability over two classes.

$$F(xi) = \frac{e^{xi}}{\sum_{j=0}^{k} xj} \quad (3)$$

Where, k=number of classes, here number of classes=2 and x is probability of input image. Then, softmax function calculates probability over two classes and final output is evaluated. The LeNet classification architecture as shown in figure 2.

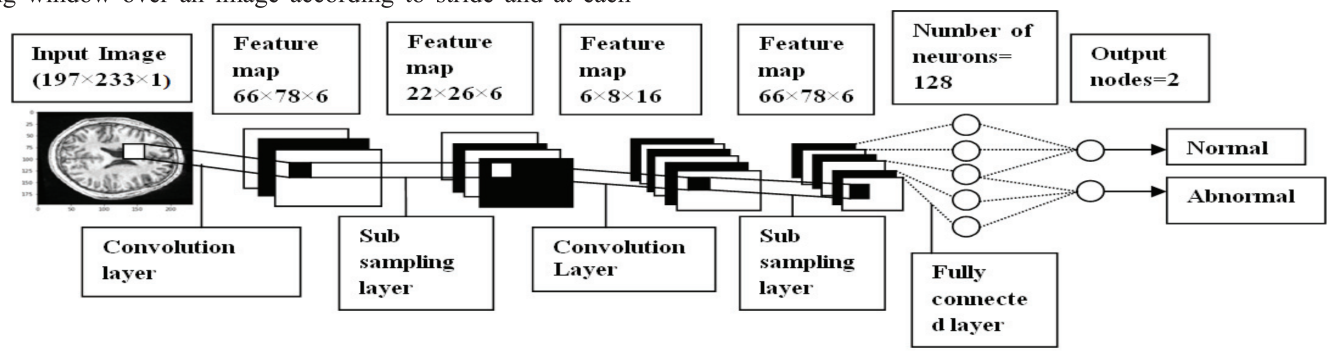


Fig. 2. LeNet Architecture

ii. Model Compiling

Once the model is created, the same is compiled using Stochastic Gradient Descent Optimizer. Stochastic gradient descent (SGD) optimizer evaluate model performance by considering any random sample, calculates slope by taking partial derivative with respect to weights and updates parameters of every sample one by one where the learning rate range between 0.0 and 1.0. Loss is calculated using binary cross entropy which minimizes loss between actual and predicted outcome. If loss is too large, the error is back propagated to the optimizer, again SGD optimize weights of every sample and evaluate model performance. This process is repeated up to number of epochs. The cross entropy and the cross entropy loss (where c=2) is calculated based on the equations (4) and (5).

$$\text{Cross entropy} = -\sum_{i}^{c} yi \log xi \quad (4)$$

In binary classification problem, where c=2 cross entropy loss is defined as:

$$\text{Cross entropy} = -y1\log(x1) - (1 - y1)\log(1 - x1) \quad (5)$$

Where, yi is one hot encoder label [0, 1] of samples and xi is output coming from fully connected layer.

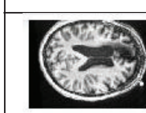
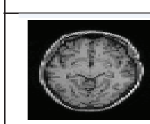
iii. Model Fitting

In training phase, data is trained using number of epochs and batch size. Epoch is one forward pass and one backward pass of all the training examples and batch size is the Total number of training examples present in a single batch. The number of epochs is selected on the basis of overfitting and underfitting problem. In this research, number of epochs used is 25 and batch size 64.after training is completed, saved the model weights using keras library.

iv. Model Testing And Prediction

Load unseen dataset apart from training dataset for testing purpose, perform pre-processing on that dataset and passed this dataset to our saved models. Model learned relationship between pixel and gave the prediction. The prediction is in terms of probability over two classes that is normal and abnormal image. We tested on 200 images. The accuracy got on training dataset is 97% and on testing dataset is 96%.Our model first classify single image and its prediction as shown in table 1.

TABLE I. .PREDICTION RESULT FOR SINGLE IMAGE CLASSIFICATION

MRI image	Probability	Prediction
	[[0.21497692,0.78502315]]	Abnormal
	[[0.61612856,0.38387135]]	Normal

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.

D. Phase II: SEGMENTATION

i. Model Building

For segmentation, SegNet[3] based architecture is used as shown in the figure 3. This network has an encoder network and decoder network followed by pixel wise classification layer. The model consists of two aspects a encoder and a decoder. Encoder part has 4 convolution with 64, 128,256,512 filters each of size 3×3 to produce set of feature maps followed by batch Normalization for normalizing data. Rectified linear unit (Relu) is applied to remove node which has negative value followed by zero padding to preserve dimension of image. Following that, 3 maxpooling layer was used with window size 2×2.while doing 2×2 maxpooling, pooling indices are stored in encoder feature map. Decoder also uses 4 convolution layer with filter size 512, 256, 128, 64 followed by BatchNormalization but decoder used 3 upsampling layer instead of maxpooling. The decoder part upsamples feature using stored pooling indices from corresponding encoder part. These feature maps was convolved over 4 convolutions to generate dense feature map. BatchNormalization was performed on each layer. Upsampling layer recalled stored map-pooling indices at corresponding encoder part and reconstruct features. Decoder in the network produces same feature map with same size and same number of channels as their encoder part. Then this high dimensional feature was passed to softmax layer for pixel-wise classification. Softmax function calculate probability of number of classes and predict segmentation corresponds to class which has maximum probability of at each pixel. The architecture of SegNet shown in figure 3.

ii. Model Compiling

After model is created, compiled using stochastic gradient descent optimizer. Stochastic gradient descent randomly selects minibatches of samples from dataset and directly minimizes testing loss and random noise present in dataset. Loss is calculated using mean squared error as given in equation (6).

$$\text{Mean squared error}=\frac{1}{N}\sum_{i=1}^{N}(yi-\bar{y}i) \tag{6}$$

Where, N=array of pixel, yi=actual probability and $\bar{yi}$= predicted probability.

iii. Model Fitting

In training phase, data is trained using number of epochs and batch size. Epoch is one forward pass and one backward pass of all the training examples and batch size is the Total number of training examples present in a single batch. The number of epochs is selected on the basis of overfitting and underfitting problem. In this research, number of epochs used is 35 and batch size 5. After training is completed, saved the model weights in two dimensional array for post processing.

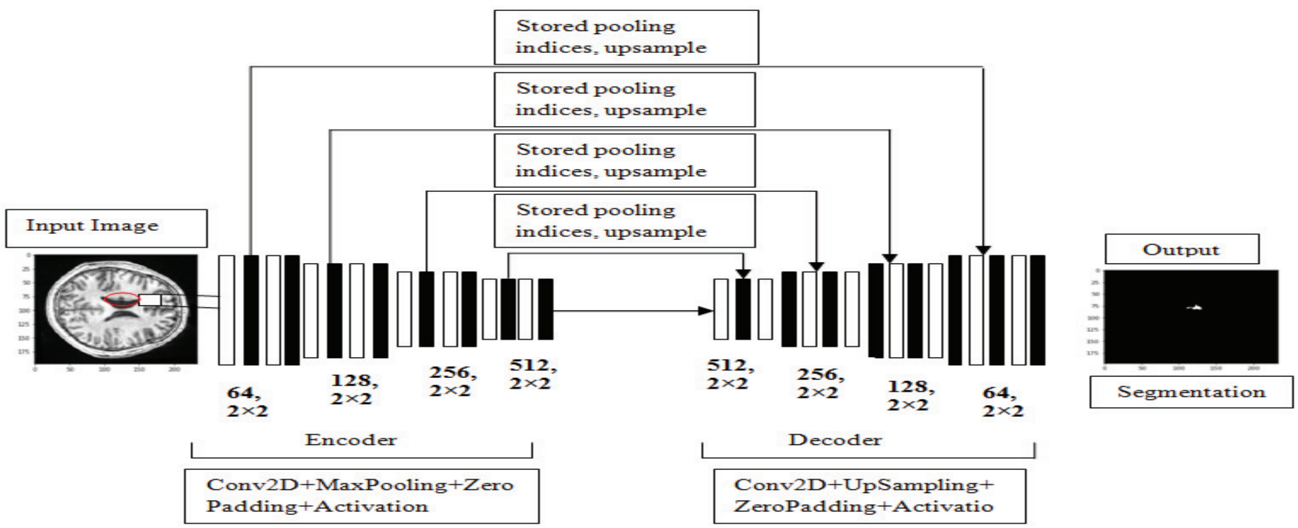


Fig. 3. Architecture of Segmentation

iv. Model Testing And Prediction

For prediction, we gave unseen brain abnormal image to the model. Again model trained weights and divided image into number of segments for performing pixel wise classification then calculate probability of each class and predict segmentation corresponds to class which has maximum probability of at each pixel. The segmentation model tested on three anatomical planes, axial, coronal and sagittal. The accuracy got on training dataset is 87% and on testing dataset is 84-85%.Model first segmented single image and result as shown in figure.4

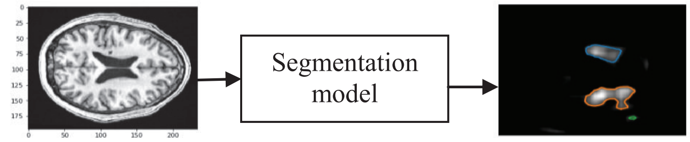


Fig. 4. Segmentation result

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.

V. EXPERIMENTS AND ANALYSIS

This section describes library used for building neural network, followed by classification and segmentation accuracy.

The dataset in Nifti format was concatenated across x and y axes, processed images using nibabel (https://nipy.org/nibabel/) and then converted into two dimensional array using numpy and pre-processed using python skimage (https://scikit-image.org/) and OpenCV (https://opencv.org/) libraries.

For the actual implementation, python is used as a main programming language. Keras (https://keras.io/) and tensorflow (https://www.tensorflow.org/) libraries are used for training neural network. Operating system used is Ubuntu 16.0.Due to unavailability of GPU, the experiments was run on CPU 8GB RAM.

A. Classification Result

The aim of classification is to classify MRI images into normal and abnormal (suffered from brain stroke).The dataset contained 229 T1-weighted MRI images suffered from stroke. After performing pre-processing, we took 210 abnormal images and 210 normal images. Then we labelled normal patient images as 0 and abnormal patient images as 1 for classification. For instance, figure 5 and 6 shows normal and abnormal patient images. Consequently, 420 images used for classification. Further, dataset is divided into two parts-70% data for training and 30% data for testing, respectively. Then passed that images and their labels to the LeNet. Before training any neural network, hyperparameter are essential to run the network. In learning phase we set value of hyperparameter. The hyperparameter is given in table 2 .Formula for calculating number of parameters learned during training is given below in equation (7).

$$O = \frac{I-K-2P}{S} + 1 \tag{7}$$

Where, I=Input image size, K=kernel size, P=padding and S=stride.

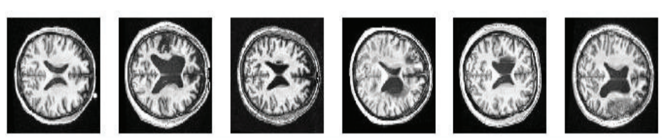


Fig. 5. .Abnormal Axial Images

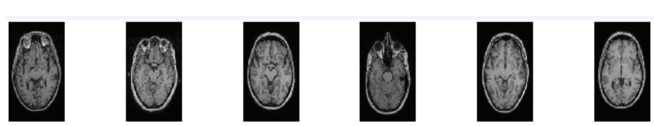


Fig. 6. Normal Axial Images

TABLE II. HYPERPARAMETERS USED IN CLASSIFICATION

Parameters	**Description**
Convolution layer	2 convolutional layer are used. -Kernel size: 5×5 -stride: 3×3
Feature map	The number of feature map used for 1st and 2nd convolution layer is 6 and 16 respectively.
Pooling layer	2 Max-pooling layer are used. Each of Pooling size: 3×3.
Nodes of fully connected layer	128
Output nodes	Output node has 2 classes(0 for normal and 1 for abnormal)
Learning rate	0.001
Optimization	Stochastic gradient descent
Batch size	64
Number of epochs	25
Dropout	0.5

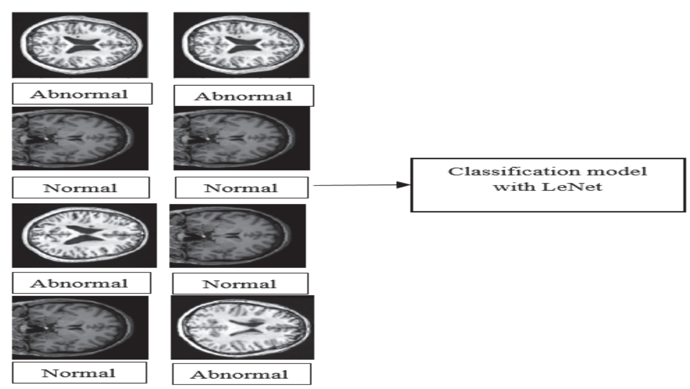


Fig. 7. Classification system using LeNet

After training and testing completed, performance was check using confusion matrix. Confusion matrix is a table use for visualizing performance of an algorithm. Diagonal elements show the appropriate classified labels and rest of elements are incorrectly classified instances. The confusion matrix of brain stroke classification is given in table 3.

TABLE III. CONFUSION MATRIX OF BRAIN STROKE CLASSIFICATION

Actual	**Predicted**	
	Abnormal	**Normal**
Abnormal	True Positive(TP)	False Positive(FP)
Normal	False Negative(FN)	True Negative(TN)

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.

TP is the number of abnormal images correctly classified, TN represents number of normal images correctly classified, FP shows number of normal images classified as abnormal and FN represents number of abnormal images classified as normal. The performance analysis of proposed method evaluated with parameters, recall, precision, f1-score and accuracy as given in equation (8), (9), (10) and (11). The confusion matrix of proposed method is given in table 4 and classification performance as shown in table 5.

$$\text{Recall}=\frac{TP}{TP+FN} \quad (8)$$

$$\text{Precision}=\frac{TP}{TP+FP} \quad (9)$$

$$\text{F1 Score}=\frac{2*recall*precision}{recall+precision} \quad (10)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

TABLE IV. CONFUSION MATRIX OF PROPOSED METHOD

Testing		**Predicted value**	
		Abnormal	**Normal**
Actual value	**Abnormal**	True Positive (TP)=67	False Positive (FP)=4
	Normal	False negative (FN)=0	True Negative(TN)=51

TABLE V. CLASSIFICATION PERFORMANCE OF PROPOSED METHOD

	Accuracy	**Precision**	**Recall**	**F1-Score**
LeNet	0.9894	0.97	0.97	0.97

The performance of model was checked by drawing training versus testing accuracy and cost function was evaluated using binary cross entropy. Figure 8 shows performance of model accuracy of brain stroke classification and figure 9 shows model loss at every iteration

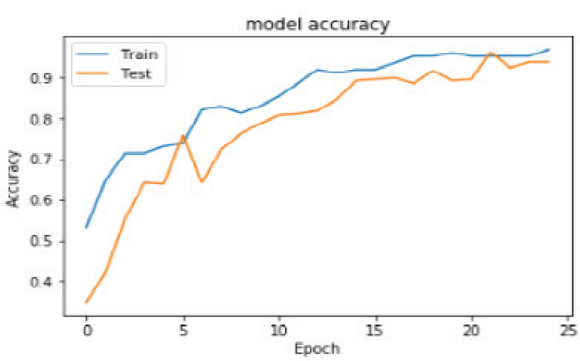


Fig. 8. Accuracy for brain stroke classification

In above diagram number of epochs used is 25 and batch size 64. As we increased number of epochs training and testing accuracy increased. Table 6 shows training and testing accuracy of brain stroke classification.

TABLE VI. TRAINING AND TESTING ACCURACY OF CLASSIFICATION

Training accuracy	97%
Testing accuracy	93%

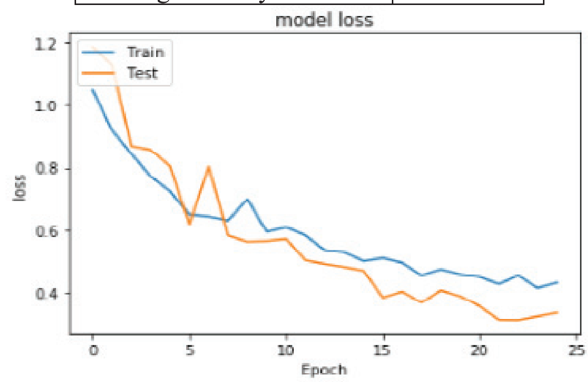


Fig. 9. Loss for brain stroke classification

The training and testing loss is calculated by taking gradient of every data points and search for local minimum value. The process is repeated up to 25 epochs. The graph shows that as we increased number of epochs loss is decreasing continuously.

B. Segmentation Result

The aim of segmentation is to delineate abnormal region of interest from image. For segmentation, we used abnormal images and their labels. For segmentation we took 210 abnormal images and 210 it's manually segmented labels and used this total 420 images from segmentation. Then same as classification, dataset is divided into training (70%) and testing (30%).The hyperparameter used in segmentation given in below table 6.For instance, figure 10 and 11 shows abnormal images and their manually segmented labels.

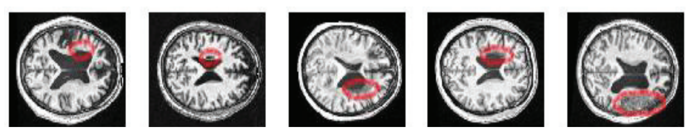


Fig. 10. Abnormal Axial Images

Fig. 11. Manually segmented label

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.

TABLE VII. HYPERPARAMETERS USED IN SEGMENTATION

Parameters	**Description**
Convolution layer	4 convolution layer are used. -Kernel size: 3×3(used for both encoder and decoder).
Feature map	The number of feature map used for 1st, 2nd, 3rd and 4th convolution layer are 64,128,256 and 512 respectively (used for both encoder and decoder).
Max-Pooling layer	3 Max-pooling layer are used. For each layer Pooling size: 3×3(used for encoder).
Up-Sampling layer	3 Up-sampling layer are used. Each layer Pooling size: 3×3(used for decoder)
Zero padding	4 zero padding layer are used and each layer. Padding size: 2×2(used for both encoder and decoder).
Learning rate	0.001
Optimization	Stochastic gradient descent
Batch size	128
Number of epochs	20
Dropout	0.5

After training and testing is completed, SegNet automatically segment out abnormal region from MRI images. Model performed segmentation on axial, coronal and sagittal plane. Then saved this all result into two dimensional array for delineation.

Loaded all this volume for delineation, took coordinate of the continuous pixel having same intensity, and draw contour along boundary to join that points. The overall accuracy got for segmentation is 91%.

The performance of model was checked by drawing training versus testing accuracy and cost function was evaluated using mean squared error. Figure 12 shows the performance of cost function for segmentation model and figure 13 shows loss of model based on number of epochs.

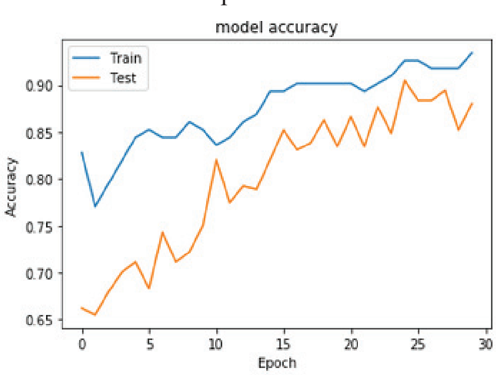


Fig. 12. Accuracy of segmentation

Above figure 12 shows that as we increased number of epochs the training and testing accuracy increased. The graph indicates that there is increasing trend in accuracy on both training and testing data.

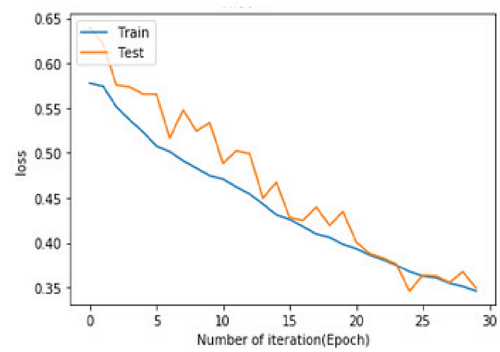


Fig. 13. Number of epoch versus loss (Segmentation)

Above figure 13 shows that value of loss change along with number of patient data. The graph indicates that there is decreasing trend of loss as the network undergoes training process.

VI. CONCLUSION AND FUTURE WORK

This research used brain stroke images for classification and segmentation. We used two types of deep learning models, LeNet for classification and auto encoder decoder for segmentation. The dataset used in this research are NIFTI format, model is tested on 406 images. Accuracy got on classification model is 96% and segmentation model is 85%.The experimental results show that these deep neural networks are absolutely relevant to brain stroke diagnosis. Therefore, the accuracy got on proposed method is better than previous methods.

Deep learning is a promising technique in medical imaging, although there are many challenges of the approach. The major challenge is caused by lack of medical images. In future, the model will build on large medical dataset, we will improve accuracy of segmentation algorithm and find exact region of brain stroke using some deep learning complex pre-trained models.

REFERENCES

[1] Greene, John, and Ian Bone,"Understanding neurology: A problem-oriented approach", *CRC Press*, 2007.

[2] Pouyanfar, Samira, et al,"A Survey on Deep Learning: Algorithms, Techniques, and Applications", *ACM Computing Surveys (CSUR)*, vol.51, no.5, pp.92, 2018.

[3] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla,"Segnet: A deep convolutional encoder-decoder architecture for image segmentation", *IEEE transactions on pattern analysis and machine intelligence,* vol.39, no.12, pp.2481-2495, 2017.

[4] Liu, Xiaolong, Zhidong Deng, and Yuhan Yang,"Recent progress in semantic image segmentation", *Artificial Intelligence Review*, pp.1-18, 2018.

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.

[5] Rajini, N. Hema, and R. Bhavani,"Computer aided detection of ischemic stroke using segmentation and texture features", *Measurement*, vol.46, no.6, pp.1865-1874, 2013.

[6] Fuk-hay Tang, Douglas K.S. Ng, and Daniel H.K. Chow, "An image feature approach for computer-aided detection of ischemic stroke", *Computers in Biology and Medicine*,vol. 41, no.8, pp. 529-536, 2011.

[7] Mazurowski, Maciej A., et al,"Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI", *Journal of Magnetic Resonance Imaging*, vol.49, no.4, pp.939-954, 2019.

[8] Yamashita, Rikiya,"Convolutional neural networks: an overview and application in radiology", *Insights into imaging*, vol.9, no.4, pp.1-19, 2018.

[9] Razzak, Muhammad Imran, Saeeda Naz, and Ahmad Zaib,"Deep learning for medical image processing: Overview, challenges and the future", *Classification in BioApps*, Springer, pp.323-350, 2018.

[10] Minar, Matiur Rahman, and Jibon Naher,"Recent Advances in Deep Learning: An Overview", *arXiv preprint arXiv: 1807.08169*, 2018.

[11] Curtis, Frank E., and Katya Scheinberg,"Optimization methods for supervised machine learning: From linear models to deep learning", *Leading Developments from INFORMS Communities*, INFORMS, pp. 89-114, 2017.

Authorized licensed use limited to: University of Wollongong. Downloaded on May 29,2020 at 23:40:20 UTC from IEEE Xplore. Restrictions apply.