



Fully Automated Magnetic Resonance Detection and Segmentation of Brain using Convolutional Neural Network

Atheel Sabih Shaker

Computer Engineering Techniques, Baghdad College of Economic Sciences University
Baghdad, Iraq.

atheel.sabih@baghdadcollege.edu.iq

Article history: Received 1, August, 2021, Accepted 26, September, 2021, Published in October 2021.

Doi: 10.30526/34.4.2710

Abstract

The brain's magnetic resonance imaging (MRI) is tasked with finding the pixels or voxels that establish where the brain is in a medical image. The Convolutional Neural Network (CNN) can process curved baselines that frequently occur in scanned documents. Next, the lines are separated into characters. The Convolutional Neural Network (CNN) can process curved baselines that frequently occur in scanned documents case of fonts with a fixed MRI width, the gaps are analyzed and split. Otherwise, a limited region above the baseline is analyzed, separated, and classified. The words with the lowest recognition score are split into further characters x until the result improves. If this does not improve the recognition score, contours are merged and classified again to check the change in the recognition score. The features for classification are extracted from small fixed-size patches over neighboring contours and matched against the trained deep learning representations. This approach enables Tesseract to easily handle MRI sample results broken into multiple parts, which is impossible if each contour is processed separately. Hard to read! Try to split sentences. The CNN inception network seems to be a suitable choice for the evaluation of the synthetic MRI samples with 3000 features, and 12000 samples of images as data augmentation capacities favors data that is similar to the original training set and thus unlikely to contain new information content with an accuracy of 98.68%. The error is only 1.32% with the increasing the number of training samples, but the most significant impact in reducing the error can be made by increasing the number of samples.

Keywords: Technological treatment, deep learning, machine learning, convolutional neural networks, magnetic resonance, brain.



1. Introduction

IT provides us with an overview of several related areas of research. Since no other algorithms have been proposed that exactly match our goal, many similar tasks are examined, especially the detection and text extraction from brains, and the detection of objects in scanned images and the classification of forms. In general, these algorithms consist of detecting brain-like objects, followed by the extraction of information in the form of human-readable text, similar to what we aim to do. The new image, which may only contain a part of the brain, is matched to the existing one, and the MRI is executed again. This multi-view approach ensures robust text extraction when occlusions or reflections are present 'as mentioned in [1]. The final output of the algorithm consists of the detected brain type, the recognized text, its position on the brain, as well as its confidence. The goal of the practical task is to create a prototype of a deep learning application that allows the user to acquire images, on which the previously listed steps to extract the content are performed. Deep learning is a machine learning method that similarly solves problems to how a human brain solves problems. This past decade it has become an x powerful instrument for solving various tasks such as speech recognition, language processing, and numerous imaging tasks. It has also opened up many possibilities for more accurate tools x such as prediction, segmentation, and analysis of medical images [2]. Deep learning methods are also relatively easy to deploy, and a deep learning architecture that is built for one task can be trained to work, as we show in **Table 1**.

Table 1: Table for predicting the training process for the architecture.

Batch Training	Yes
Batch Normalization	Scalable
Batch Size	Moderate

Deep learning will learn how to determine the brain masks based on a training set of already segmented brain masks with little tuning required. Ren et al. [3] introduce a method for brain stripping that uses a fully convolutional neural network. Their method is compared with six existing brain stripping methods on three different data sets. It performs better than all of the conventional methods in some metrics also says that its method should work well on other medical image modalities in contrast to the existing methods.

All of the patients in the data set have brain tumors. The existing solutions that are not based on deep learning do not handle this well. Therefore, it would be interesting to see if deep learning can be used to tackle this problem. In [4] they showed that their deep learning Provided method was better compared to existing brain stripping methods at handling images of patients with tumors in their data set. Consequently, this research aims to explore if deep learning methods can handle the tumor data from the data set adequately.

There are Varying amounts of data in the three data sets used in this paper. There are many more MRI images in the data from the hospital compared to the other two data sets. This unbalanced composition of the data sets can cause problems for deep learning methods. It can cause problems such as the deep learning methods being

very good at predicting data from the most extensive data set but poorer on data sets with the images. Magnetic Resonance Imaging (MRI) is an imaging technique that is regularly used for taking images used for medical analysis. The images are obtained by using a strong magnetic field to align hydrogen atoms inside the body. Radiofrequency energy from the machine is then used to excite the hydrogen atoms [5]. After the machine stops emitting Radiofrequency energy, the hydrogen atoms return to their resting state, causing the atoms to emit energy. The energy is then read by antennas inside the MRI machine. As we show in **Figure 1**, we also show an example of a simple neural network with three layers in **Figure 2**.

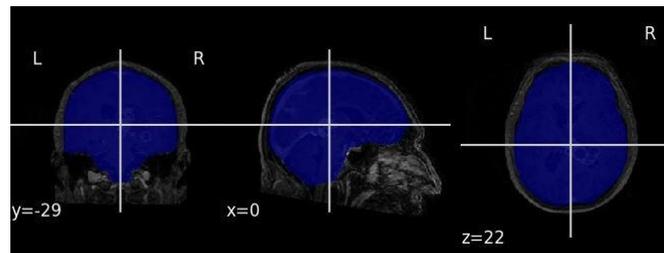


Figure 1. Blue marks where the brain is in the MRI scan
 Input layer Hidden layer Output layer

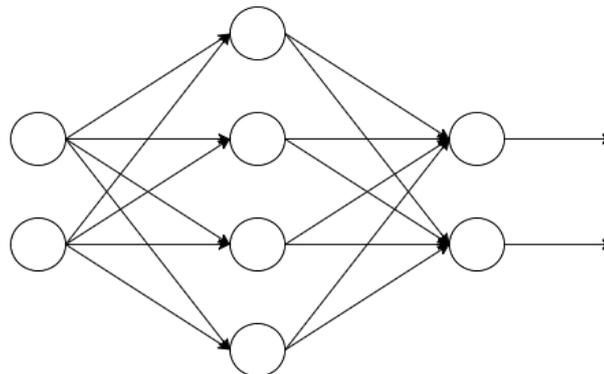


Figure 2. Example of a simple neural network with three layers.

1.1.Purpose of Study

The overall goal for this study will be to explore further deep learning applied to MRI of brain stripping, especially on data from the hospitals. Another goal for this paper will be to examine how different hardware configurations impact the training time of different CNNs [6]. Also, this paper will test if these deep learning methods can be used to do segmentation. Based on the problem description in the previous section and the goals, the following Research Question (RQ) have been defined and will be addressed in this paper:

- Q 1. How important are different deep learning architectures to train on data from the same source before performing brain stripping?
- Q 2. How important is the balance between the amount of data in each data set for deep learning architectures when doing brain stripping?
- Q 3. Is it important for deep learning architectures to have MRI scans with the same voxel

size when performing brain stripping?

$$\sigma(z) = \max(0, z) \quad (1)$$

With equation (1) the neuron outputs its computed value if it is over zero; otherwise, it outputs zero. A good and non-biased segmentation method for brain stripping is, therefore, a very valuable tool.

A neural network is trained by giving it example cases and their corresponding solutions. The network computes the prediction and the loss for the example cases.

1.2. Back Training

The network can update its parameter after seeing more than one example. This is called batch training. When batch training is used, the network is fed a batch of example cases; the loss is computed and averaged for the batch. The number of examples fed into the network is called the batch size. Since the loss is averaged over a set of example cases, it becomes a more accurate estimation of how close the network's function is to the problem's underlying function instead of just one example case. The updates to the parameters are therefore more stable. Consequently, the network can learn more smoothly [7]. A more significant learning rate can also be used when larger batch size is used.

1.3. Optimizers

There are different alternatives to plain backpropagation for training neural networks. Ling, H [8], provides an overview of some of the alternatives. Using momentum, the updates can be estimated to be larger or lower for the parameters for each update step. It does this by adding a fraction of the last update to the current update value. It approximates future values of the parameters to be updated. Then it uses this to update the parameters.

1.4. Aim of Study

A system for real-time brain recognition using deep learning-based CNN technique. The study is designed and implemented a complete real-time magnetic resonance system based on deep learning. The system broadcasts the location and MR detection of the brain through a deep learning based approach and can be used by other applications. An inexpensive way to generate accurate labeled data using CNN for magnetic resonance samples. We use deep learning to easily generate labeled data using the aligned RGB and depth images. We found this expensively and through deep learning, developers can generate their customized system without difficulty. Another advantage of our approach is that the system uses actual raw depth MR images as training samples. These naturally capture realistic noise such as shadows and hardware noise. Using computer-generated graphics are very difficult to simulate these noisy effects. Note that the end-users do need to use deep learning; they are only used in training. A computational insight about the convolutional neural network (CNN) based data model. To the best of our knowledge, there seems to be literature comparing SVM and random forest from a computational perspective. We provide an in-depth complexity analysis of the two methods rather than merely reporting experimental accuracy as done in most machine learning literature Extensive experimental evaluations of the system We conduct extensive experiments evaluating the effectiveness of the random

forest classifier by systematically exploring a large space of parameters. Interesting results lead to a deeper understanding of CNN.

2.Related Work

Researchers in [9] propose an algorithm for rectangular segment objects that may overlap and are placed on a lightly textured background with unknown color, which is the usual output of scanner preview images. The authors in [10] aim at detecting the background color, which is challenging when the image consists primarily of differently colored foreground objects, but it allows them to segment the objects more easily. To retrieve the background color, they first separate the image into line segments with the same tinby calculating the neighboring color differences. Under the assumption that the background color segments are long, a voting scheme is used to extract possible background colors. For each background color candidate, the edge strength at the boundaries to non-background colors is calculated. The authors observed that the gradient between background and object colors is larger than it is between foreground colors and that the number of edge pixels correlates with the number of foreground objects. Therefore, they determine the true background color by only using line segments that exceed a certain length, are of the most frequent color, and have many edge points with high gradient values, as mentioned in [11]. Connected components are extracted and used to fit lines from these edge pixels, with weights derived from the edge strength. Neighboring pixels located in the direction of the line are added until a stopping criterion is reached Orthogonal lines are used to calculate the corners of the objects and the line segments that form the rectangle. To eliminate wrong line segments, the median and mean color differences between foreground and background along the line segments are calculated and used to derive the parameters of a score function from many automatically generated images. Next, for each rectangle candidate, the same values used for line fitting and a score that relates to the number of background pixels in the rectangle are input into a Support Vector Machine (SVM) classifier, which determines if the rectangle candidate is a proper object or not. To find better matches, the rectangle candidates are also shifted in the local neighborhood. A candidate is accepted when at most 10% of the pixels inside the rectangle are classified as background.

In summary, all approaches are specifically designed to be fast and detect potentially imperfect rectangular objects with possible overlap or missing corners. However, they rely on the uniform background color, which also must be distinct from the majority of the content found in the foreground objects. Our application usually does not satisfy both conditions, as brains are mounted on arbitrary background, and unknown lighting conditions may cause significant brightness changes. Also, the color of the brain is not necessarily different from the background. Hence, the only way of distinguishing them from the background is by the border of the brain

3.Methodology

An efficient rectangle detection system is intended to be used on mobile devices. First, they extract the edge map using the Canny edge detector. The threshold parameters are automatically selected. To improve the robustness and increase speed, all edges produced by text are removed. Consequently, a lower number of lines need to be processed. They do this by extracting connected components and calculating the bounding box for each one. The

aspect ratio, height of the bounding box, and a number of pixels about the bounding area are used as filtering criteria. Next, the filtered image is used to extract lines by applying the Hough transform. Suppose their angle does not differ more than fourteen degrees, two lines from a line bundle, accounting for some perspective distortion. The authors then group line bundles to rectangle hypotheses if all line intersections are inside the image. Finally, they compute the edge support on the dilated edge image, where the most dominant rectangle hypothesis with the highest edge support is the resulting rectangle. The edge map is dilated to increase robustness against camera distortions, causing lines that are not completely straight.

A more generic approach to extracting rectangles using a windowed Hough transform is proposed. They use a ring-shaped sliding window, where the outer diameter approximately equals the size of the largest and the inner diameter the size of the smallest rectangle that can be detected. Next, the Hough transform of the region under the sliding window is calculated, where the discretization depends on the outer ring diameter. To detect the peaks in a robust manner, a modified butterfly evaluator is applied to the accumulator. To retrieve the lines, the accumulator is PNSR, as shown in equation (1) down below.

$$PNSR = \frac{\max_{i,t} |MRI|_{model}}{noise_{clean+test}} \quad (1)$$

Several schematic drawings are also frequently present, which results in a large number of lines that cannot be easily removed beforehand. With the examined approaches, grouping and checking the hypotheses is time-consuming. The printed lines inside the brain may also exhibit stronger edge characteristics than the actual separation between the brain and the background, thus leading to wrong detection results.

3.1 Software Implementation

An approach that employs deep learning approaches and applies to forms without tables or tabular structures is developed in MATLAB. They use different primitives that occur on forms, such as text, lines, vertical distances between adjacent rows, intents (positive and negative), and nine types of corners encoded into a string by iterating over the objects that are sorted by their vertical position. The number of common tokens of the learned blank and the new incoming form is counted with and without the text blocks and combined into a score to match a new document.

3.2. Material and Methods

A convolutional neural network consists of many layers, where each layer is trained on a different subset of the training set. The elements in the subsets are generated by bagging, which chooses elements randomly and with restitution. Therefore, a sample may appear multiple times. When training the layers, a random subset of the features in the feature vector is used to split the samples for each node. In our case, the number of features used for the splits is half the square root of all features. From all calculated splits, the one that causes the most significant decrease of entropy and, therefore, the largest information gain of the label histogram is used. In our case, the learning is completed if CNN-Layers are generated or the estimated classification error is smaller than 1%. The error is estimated during training by classifying samples that were not chosen by the bagging procedure. The layers in the

convolutional neural network are not pruned. For classification, each tree generates a result from the input data, and a majority vote is used to derive the final class label.

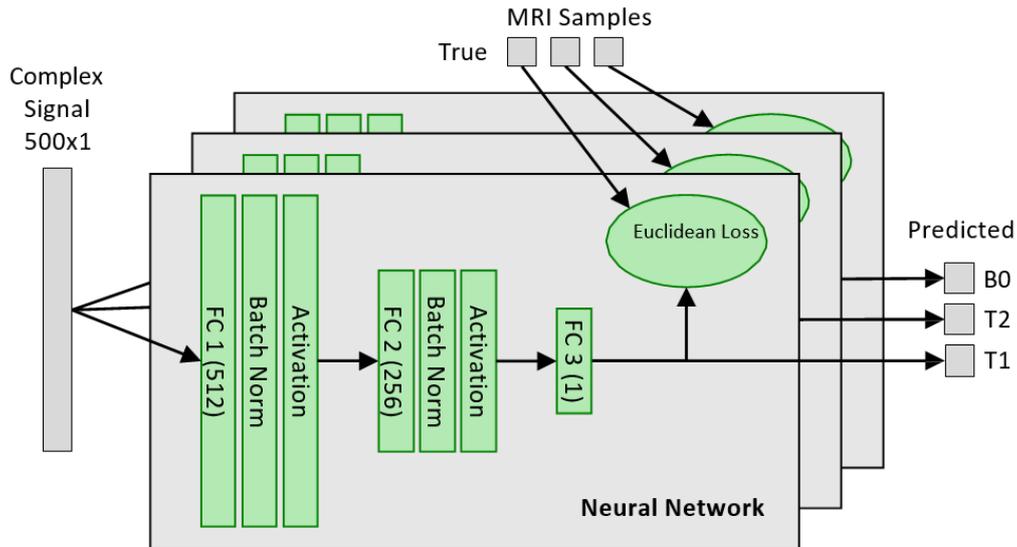


Figure 3: Architecture of neural network approach for the segmentation of MRI samples.

It is shown in **Figure 3** That architecture of the neural network approach for the segmentation of MRI samples we use convolutional neural networks because they have several advantages. We state that increasing the number of layers does not cause overfitting. Therefore, a large number can be used safely, as it increases accuracy. However, with an increasing number of layers, the improvement in classification accuracy declines at a certain point. Furthermore, a larger number of layers increases the time needed for training and linearly increases the time for classification. Training and classification are also speedy since each tree can be processed in parallel. Finally, a very important factor is classification performance. Supervised deep learning compares some popular approaches, including support vector machines, boosted layers, convolutional neural networks, and others. The dataset contains MRI sample for training and testing the automated system. The research concludes that there exists no universally superior algorithm, as no one excelled at each problem. Their test results show that calibrated boosted layers, convolutional neural networks, and bagged layers generally deliver the best performance. The dataset was achieved from an open-source repository; the link is given by: <https://openfmri.org>

3.3. Training of System

We evaluate our use of convolutional neural networks. The research employs maximally stable extremal regions to locate brains. The brains are located by searching for some smaller areas placed within a larger one. In CNN, most of these systems are designed to detect text in natural images. Since our brain images do not contain nearly as much clutter as natural images, such sophisticated approaches like MRI are not needed. Instead, we employ a similar MRI and grouping-based approach to extract MRI from samples for brain detection. We have the extracted up-right brain image and a list that contains the position, content type, and name of the areas that should be extracted. Since the brain is not

always extracted perfectly, we need to adjust the given areas, to align with the brain's content.

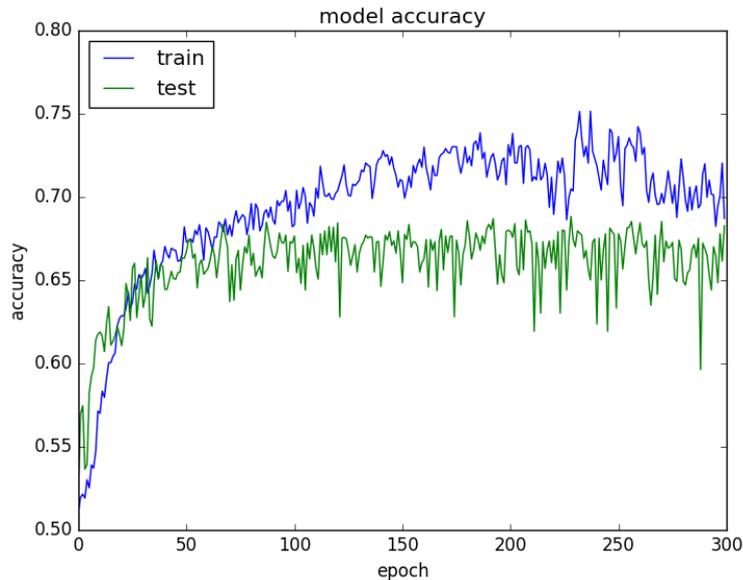


Figure 4: The training and testing graph for neural network model with dropout.

So in **Figure 4**, we show the training and testing graph for neural network model with dropout, images tend to portray similar features at different spatial locations. Accordingly, units with receptive fields at different locations will have the same weight vectors to extract similar features, known as parameter sharing and further reduces the dimensionality of the problem.

3.4. Baseline Parameters for MRI

The output of the first stage is an image of the same dimensions as the input image that contains the stroke width for each pixel. First, they compute the edge map of the input image using the Canny edge detector and initialize all the stroke width result values to infinity. For each returned edge pixel, the gradient direction is followed until another edge pixel is found. Suppose the gradient direction of the second edge pixel roughly points back to the starting pixel. In that case, all stroke width values on the line connecting them are set to the distance between the starting and the end pixel, but only if they do not already have a lower value. When all edge pixels are processed, all start pixels where an end pixel was found are revisited. The stroke width values on the lines connecting them are set to the median value of all pixels on the line. The research note that this is necessary to get correct results in corners, Next, they extract connected components, where neighboring pixels are assigned to the same contour if their stroke widths are similar. These contours are filtered to retrieve text regions. A contour is discarded if the stroke width variance of its enclosed pixels is too large or when its aspect ratio or diameter to stroke width is not within a threshold. Furthermore, the bounding box of a contour must be appropriately sized and not contain more than two inner contours. The resulting letters are grouped into pairs if they have a similar stroke width, height, and color and are not located too far apart. Finally, the

brain pairs are merged into text lines, and word boundaries are detected using a histogram of the brain distances.

4.Results

The final grouped regions are obtained by applying the previous clustering steps again to the co-occurrence matrix. A CNN classifier is used to prune regions that do not have a brain-shaped object. Finally, a CNN classifier is used to remove clustered regions that do not contain text. Regions for the class and classifier determine whether the input region is a brain or not, as we show in **Figure 5** in three-stage input and segmentation MRI scan. The detected text regions are grouped into brains by incorporating knowledge of the layout.

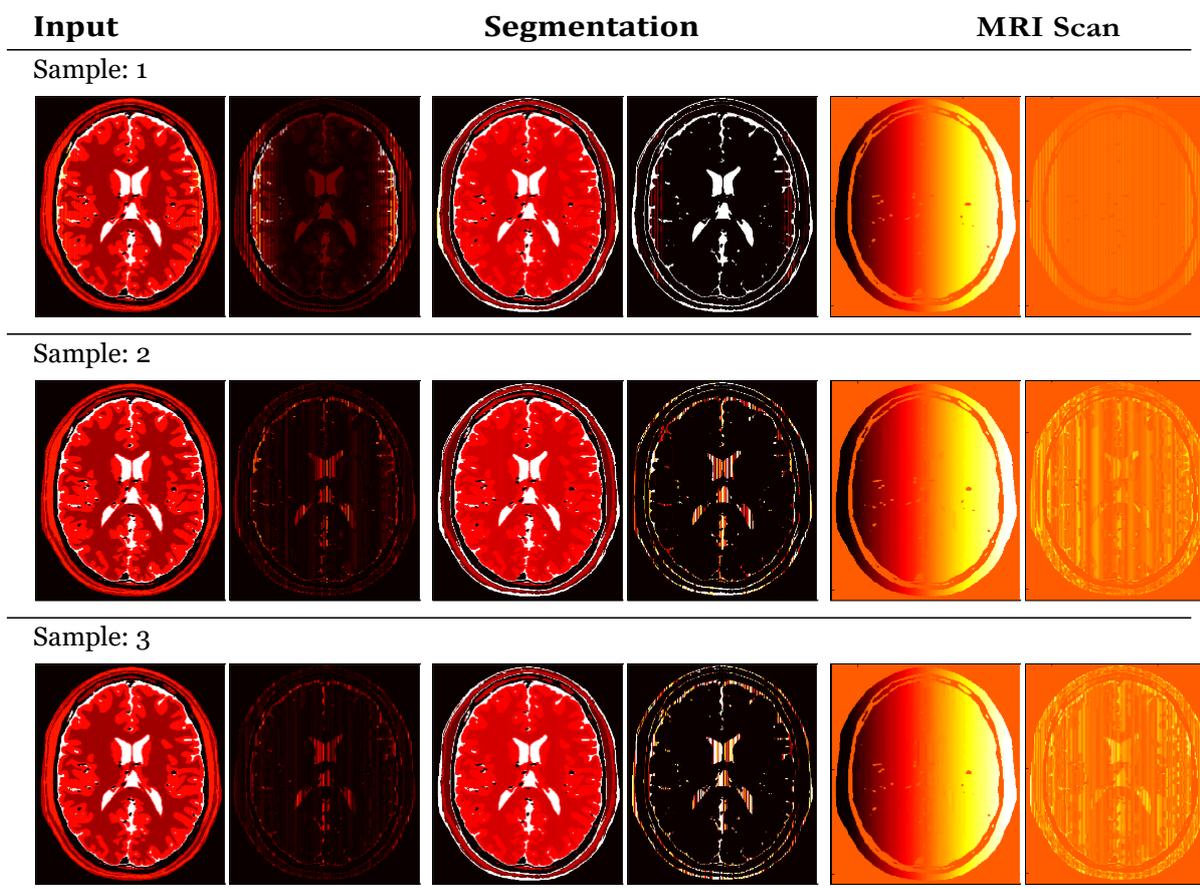


Figure 5: The three-phase processing of MRI samples from input, segmentation to the scan.

The opened image is subtracted pixel-wise from the original image. Since the brain width is small about the structuring element, they remain intact. The graph illustrates this for the first sample region. As the lines are not leveled, some parts of them stay for MRI. However, using the same approach to remove vertical lines would also remove most of the text. To prevent this, the presence and location of borders are determined first, and only those regions are filtered. At the first 20% of the left and right sides of the image, peaks are detected. A peak is reported if 70% of all pixels in a column are black. Since we assume that the text is roughly centered, the distance of the first detected peak on either side to the image border should also be approximately the same. Again, the

opening result is subtracted pixel-wise from the original image. Since the removed lines usually are not perfectly horizontal or vertical, some debris remains. We show **Table 2** Results comparing the Neural Network and empirical residual models. Cleaning and testing values are computed by comparing the predicted values to the MRI scans found using the baseline dictionary neural network method. Also, we show table 3 improvements at all Neural Network levels when the MRI rotation is included with and without clean/test. In fact, the MRI scan map image quality for the Neural Network PSNR level 2.5 is relatively close to both the empirical.

Table 2: Results comparing the Neural Network and empirical residual models. Cleaning and testing values are computed by comparing the predicted values to the MRI scans found using the baseline dictionary neural network method.

Synthetic model	Clean	Test	Clean	Test
Clean signal	0.002	0.805	0.002	0.148
Neural Network PSNR 60	0.009	0.896	0.004	0.157
Neural Network PSNR 40	0.003	0.937	0.006	0.161
Neural Network PSNR 20	0.005	1.039	0.013	0.215
Neural Network PSNR 10	0.013	1.109	0.026	0.270
Neural Network PSNR 5	0.027	1.143	0.050	0.317
Neural Network PSNR 2.5	0.077	0.923	0.093	0.378
Neural Network PSNR 1.25	0.223	0.731	0.165	0.186
Neural Network PSNR 0.625	0.492	0.642	0.266	0.209
Empirical residual model [7]	0.058	0.126	0.026	0.071

Table 3: We show improvements at all Neural Network levels when the MRI rotation is included with and without clean/test. In fact, the MRI scan map image quality for the Neural Network PSNR level 2.5 is relatively close to both the empirical.

Synthetic model	Without Clean	With Clean	Without Test	With Test
Clean	0.805	0.556	0.148	0.322
Neural Network PSNR 20	1.039	0.554	0.215	0.181
Neural Network PSNR 10	1.109	0.427	0.270	0.201
Neural Network PSNR 5	1.143	0.272	0.317	0.161
Neural Network PSNR 2.5	0.923	0.168	0.378	0.119
Neural Network PSNR 1.25	0.731	0.191	0.186	0.069
Neural Network PSNR 0.625	0.642	0.243	0.209	0.113
Empirical residual model [3]	0.902	0.126	0.148	0.071

5. Discussion

The detection of the first and last brains of the line is checked against the median bounding box contour height. If the height difference to the median contour height is larger than 25%, the bounding box does not overlap with the bounding box of the neighboring contour, and the bounding box width is smaller than 33% of the text region width, the contour is removed. At this point, the preprocessing is complete. However, the brains only consist of borders, as all processing was done on an image created by MRI the gradient magnitude. Such images cannot be processed by the Tesseract MRI. To retrieve filled brains, the original unmodified input image is adaptively MRI with block size and the filtered image

is morphologically opened with a 20×15 ellipsoidal structuring element. The large black region is caused by the rust stain. The MRI result for the second regions, also contains some clutter. When copying from the adaptively MRI image, the opened image is used as a mask, as mentioned in [12]. The results for both regions that are passed to Tesseract are everything that does not belong to a removed brain.

Sample of the brain image at the top right, and a new input image containing the right part of the brain at a larger magnification. Further, the extracted key points, the calculated matches between the images, and the location of the new image in relation to the reference brain are displayed. Each completely visible region is warped into its upright position, where the size is approximately the same as in the new input image. Since each region is processed and warped separately, the process can be executed in parallel, and fewer resources are needed since not the entire image must be processed. The preprocessing and MRI steps, as described earlier, are executed for each new region. The results are merged with the existing ones, where for each region only the one with the highest MRI confidence is kept.

6. Conclusion

In this novel research work, we have developed an intelligence expert system for the detection of brain MR using convolutional neural network. Deep learning methods demonstrated good detection and accuracy. For the training, testing, and validation of an Open-MRI-based dataset for brain diseases; well-known MATLAB R2018b software was used for this purpose. A convolutional neural network is written with one hidden layer, 16 input neurons and two outputs either healthy or not. The data are split into train and test datasets with 70% for training, 15% validation, and 15% for testing. Accuracy is found to vary between 91-92% depending on the number of iteration or epochs. Moreover, the CNN method can work well with, varying number of features. Generally, its performance for detection purposes does not suffer from extra features, meaning that in, real-world environment they offer the possibility of using all existing data features. It was also shown that there may not be a need to create different models for brain disease detection and that one model trained on data can detect brain disease based on MR samples efficiently. The CNN inception network seems to be a suitable choice for the evaluation of the synthetic MRI samples with 3000 features and 12000 samples of images as data augmentation capacities favor data that is similar to the original training set and thus unlikely to contain new information content with an accuracy of 98.68%. The error is only 1.32% with the increasing the number of training samples, but the most significant impact in reducing the error can be made by increasing the number of samples.

References

1. Yan, J.; Lim, J. C.-S.; Townsend, D. W. MRI-guided brain pet image filtering and partial volume correction, *Physics in medicine and biology*, **2015**, *60*, 3, 961,
2. Ronneberger, O.; Fischer, P.; Brox, T.U-net: Convolutional networks for biomedical image segmentation, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, **2015**, 234–241.
3. Ren, S.; Girshick, K. He, R. et al. Faster r-cnn: Towards real-time object detection with region proposal networks, in *Advances in neural information processing*

systems, **2015**, pp. 91–99.

4. Dong, C.; Loy, C. C.; K. He et al., Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, **2016**, 38, 2, 295–307.
5. Wang, S.; Z. Su, L. Ying et al., Accelerating magnetic resonance imaging via deep learning, in *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on. IEEE*, **2016**, 514–517.
6. Kang, E.; Min, J.; Ye, J. C. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction, *arXiv preprint arXiv:1610.09736*, 2016.
7. Wolterink, J. M.; Leiner, T.; Viergever M. A. et al., Generative adversarial networks for noise reduction in low-dose ct, *IEEE Transactions on Medical Imaging*, **2017**.
8. Johnson, J.; Alahi, A. ; Fei-Fei, L. Perceptual losses for realtime style transfer and super-resolution, in *European Conference on Computer Vision. Springer*, **2016**, 694–711.
9. Ledig, C.; Theis, L.; Huszar, F. et al., Photo-realistic single image superresolution using a generative adversarial network, *arXiv preprint*, **2016**.
10. Yang, Q.; Yan, P.; Kalra, M. K. et al., Ct image denoising with perceptive deep neural networks,” *arXiv preprint arXiv:1702.07019*, **2017**.
11. Han, Y. S. ; Yoo, J. ; Ye, J. C. Deep learning with domain adaptation for accelerated projection reconstruction mr, *arXiv preprint arXiv:1703.01135*, **2017**.
12. Oktay, O.; Ferrante, E.; Kamnitsas, K. et al., Anatomically constrained neural networks (acnn): application to cardiac image enhancement and segmentation, *IEEE transactions on medical imaging*, **2017**.