Journal of Integrated and Advanced Engineering (JIAE)
Vol. 4, No. 1, March 2024: 77-86
http://asasijournal.id/index.php/jiae
http://doi.org/10.51662/jiae.v4i1.144

Identification of Alzheimer's disease using Convolutional Neural Network

Cedric Obundaa Nnah*[1], Yu-Dong Zhang[2]
[1]Nigerian Army Training Centre (NATRAC), Nigeria
[2]Department of Informatics, University of Leicester, United of Kingdom

Abstract
Alzheimer's disease (AD) as a brain disease has caused a progressive, devastating effect on the memory and general mental and physical coordination of victims. The impact on victims is irreversible, and the cause has yet to be identified. The treatment at full-blown can be difficult, but it could be properly managed in the early phase. Hence, there is a need for an efficient and effective early diagnosis. Machine learning techniques have proved to be successful in image classification. It was on this premise that this paper adopted a machine learning approach. The approach used a convolutional neural network with transfer learning to classify structural Magnetic Resonance Images (sMRI) into a multi-classification of 3 classes. The classes were Normal Cognitive (NC), Mild Cognitive Impairment (MCI) and Alzheimer's Disease (AD). K-fold cross validation was employed to validate the test set. The sMRI subjects included 97 NC, 57 MCI, and 24 AD patients. The proposed method achieved an overall accuracy of 94% on classification based on the multiclass classification.

Keywords:
Alzheimer's disease;
Convolutional Neural Network;
Magnetic Resonance Images (sMRI);
Mild Cognitive Impairment;
Normal Cognitive;

Article History:
Received: January 1, 2024
Revised: March 2, 2024
Accepted: March 4, 2024
Published: March 12, 2024

Corresponding Author:
Cedric Obundaa Nnah
Nigerian Army Training Centre (NATRAC), Nigeria
Email:

INTRODUCTION

Alzheimer's disease is a devastating brain disease that develops gradually, starting from a part of the brain and extending to other parts over time. It results from unusual protein representations, which causes "nerve cells" to die and thereafter leads to brain tissue loss [1]. The damage done to the brain is actually a physical one [1]. The effect of this is that the affected individual begins to have a "poorly recognized failure of memory and slowly becomes severe" [2]. This means that the individual suffering from this kind of disease begins to lose the memory of recent events and, over a period of time, of events that happened in the distant past [1]. Other conditions that characterize the disease as it progresses include difficulty in speech, poor vision, lack of coordination, and reasoning [2]. A phase exists when the symptoms begin to develop gradually with "problems in cognition, learning and memory" [3]. At this stage, it can actually be observed but does not prevent the individual from "carrying out everyday activities"; it is called MCI [3]. The risk factors for this disease include age, gender, lifestyle, and health problems.

Diagnosing Alzheimer's disease in a patient requires the patient to make a visit to the physician and report observed symptoms. The doctor, however, will have to conduct a series of medical examinations to determine the presence of dementia and the level of impact it has on the patient [4]. These examinations include physical, blood, and brain image tests to ascertain the level of memory loss and the individual's overall coordination. Of the various

tests conducted, brain images have been predominantly used to detect "visible abnormalities" in relation to AD [1]. Identifying the presence of this disease medically has been done using brain scanning techniques such as computerized tomography (CT), MRI, and Positron Emission Tomography (PET) [1]. The above methods can be cumbersome; hence, machine learning can be applied for improved identification.

Machine learning is a fast-evolving field in computer science. Machine learning automatically observes and learns the patterns in a data (image) fed into. Thereafter, it uses the knowledge and experience from that data to make predictions and decisions on future data [5]. Deep learning, an improved form of machine learning, has significantly impacted image recognition [6]. Hence, it was the approach used for this research. Machine learning methods are chiefly divided into two, which are supervised and unsupervised learning [5].

In this type of machine learning, the system applies the knowledge acquired from previous or past data "using labeled examples to predict" what the future outcome would be. The system learns from a dataset and then makes predictions on the possible output. After adequate training, the system can also provide results for fresh or additional input. Its output can be compared with the model output to ensure correctness and identify observed errors to improve the system [5].

Unsupervised machine learning is used when the instruction for training the system is unclassified and not labeled. Its main focus is on how to make deductions from a given instruction to describe an inherent pattern from an "unlabeled data". In this form, there is no comparison to determine the ideal output, but the use of given datasets is used to make deductions and predict an output of inherent figures or images from "unlabeled data" [5].

Dataset

Prof Yu-Dong Zhang provided the dataset used to implement this project at the University of Leicester. The dataset was saved in an a.mat extension file format. This is a binary format that is used in MATLAB to save files of different data. The dataset contained 178 sMRI grey images with 0, 0.5 and 1 values. The images were that of brain samples. The data set contained two files, namely "Input_Z_80.mat" which was the input data and "Target.mat" which was the label. The dataset was quite small for a machine learning problem. This was as a result of the fact that medical samples were usually difficult to come by. The reason is that it usually involves using live humans to process medical imagery to extract sample datasets and, as such, requires the consent of those involved. Several images from the dataset are presented in Figure 1.

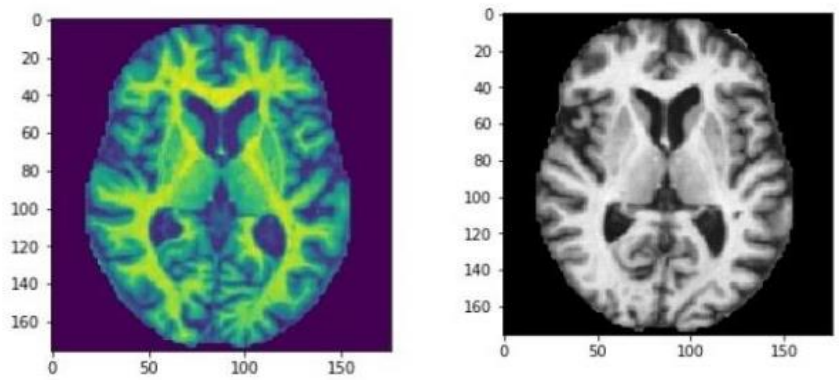


Figure 1. Images from the dataset

Label distribution

The target labels were of three classes, which suggested this was a multiclass classification problem. The classes were NC, MCI and AD. Specifically, it was a single label classification task, meaning that one of the images belonged to a particular class at any point. It is similar to binary classification, as was used by [7], but it involves comparisons among the three labels. The research used a single classifier against [8] that employed a multistage classifier for a similar task. To identify the number of images in a class, a loop through indexed label array was performed.

Vectorization

Vectorization is the process of turning the input data into tensors [9]. One-hot encoding was used to perform this operation. In this case, it was used to turn the target labels to tensors of float 64. This was because the input data should be tensors of floating point in order to be processed by the convolutional neural network. The one-hot encoding transforms the values in the vector to an all zero value, assigning only one to the index of the specific class. This transformation improves the computational performance of the network.

Balancing dataset

In the dataset used for this research, there were three classes of NC 97, MCI 57 and AD 24 imbalanced classes. The imbalanced classes could lead to inaccurate classification. In order to mitigate this, a balancing technique was employed. This was to ensure that the weight of each class was adjusted during the training so that all classes could carry similar weights [10]. K-fold cross validation was used for the validation.

METHOD

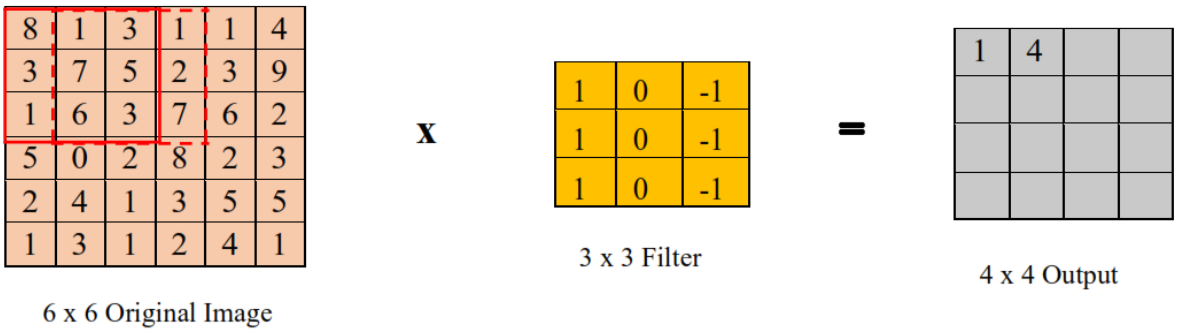
Convolutional neural network

A convolutional neural network is a form of machine learning approach based on "deep neural network" [11]. It has recently achieved success in "machine learning applications" such as "object detection, image classification and face recognition" [12]. A convolutional neural network is comprised of basically three layers. The convolutional layer is made up of a single or multiple convolutions, the pooling or sub-sampling layers and the fully connected layers [13].

Convolutional layer

The convolutional layer is usually the main starting point in a convolutional neural network [14]. In this layer, features are extracted "from images" and fed into the network. This is done with the aid of a filter smaller than the size of the original image; it reduces the image as well as ensures "relationships between pixels" are not lost [12]. For instance, if a convolution is applied to a [5x5] image using a filter of [2x2] with [1x1] stride, the output will be a [4x4] image [12]. This means that filters are applied to the image in this layer, which strides through the image. This continues until the filter reaches the image's end [15]. Figure 2 shows a simple convolution process.

=



Element-wise product and sum operation of the original image and filter

8 x 1	**1 x 0**	**3 x -1**
3 x 1	**7 x 0**	**5 x -1**
1 x 1	**6 x 0**	**3 x -1**

(8 + 0 - 3) + (3 + 0 - 5) + (1 + 0 - 3) = 1

1 x 1	**3 x 0**	**1 x -1**
7 x 1	**5 x 0**	**2 x -1**
6 x 1	**3 x 0**	**7 x -1**

(1 + 0 -1) + (7 + 0 - 2) + (6 + 0 - 7) = 4

Figure 2. Convolution process

In Figure 2, we see how a convolution is applied. Here, a [3x3] filter was used on an image size of [6x6] and a resulting output of [4x4] image was generated. When the filter strides to the edge of the image, it cannot move further. As a result, information contained at the end of the pixel would be lost [15]. If this situation had occurred at every convolution, vital information about the whole image would have been lost. To resolve this, the process of padding was applied. This is a process whereby zeros are added (padded) around the edges of the image. This ensures that all the information about the image is captured even as the filter moves to the borders of the image, thereby generating an image size that is the same as the input [15]. Figure 3 shows a picture of the padding process.

In Figure 3, it can be clearly seen how the image was padded so that the corresponding filter can learn details about the edges. [15] Explains that several of these filters can be applied to an image to enable it to learn several aspects of the image. Each of these filters generates an output that is then stacked together.

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	8	1	3	1	1	4	**0**
0	3	7	5	2	3	9	**0**
0	1	6	3	7	6	2	**0**
0	5	0	2	8	2	3	**0**
0	2	4	1	3	5	5	**0**
0	1	3	1	2	4	1	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Figure 3. A padded pixel

Pooling Layer

The pooling layers are found between a pair of convolutional layers [12]. Its major work is the reduction of the feature map, which ensures that "information between features" is adequately preserved [14]. This helps in controlling over-fitting [12]. The pooling layer has two types: Max pooling and Average pooling [15]. Max pooling, applied after a convolutional layer, is aimed at down-sampling the output received from the convolutional layer [16]. It does this by reducing the image dimension, giving room for deductions about the feature representations [16]. As a result of this reduction and down sampling of features, it invariably controls over-fitting as well as "computational cost" making the model train and learn quickly [16]. On the other hand, average pooling takes the average of all the values in the filter size. An illustration of Max pooling is presented in Figure 4.

The maximum values of each filter are extracted, which means it is the most prominent feature in the filter. Picking the Max values down samples the entire size of the image.

Flatten Layer

The flattened layer in a convolutional neural network performs a very vital function. It is usually positioned between the pooling layer and the fully connected layer. Its role is to remove all the tensor dimensions and leave just one, indicating that the tensor has been flattened to a one-dimensional vector. This process is known as flattening, where n-dimensional tensors from the pooling layer are transformed to one dimensional just before it is being fed into the fully connected layer. Figure 5 represents the result of flattening in a convolutional neural network

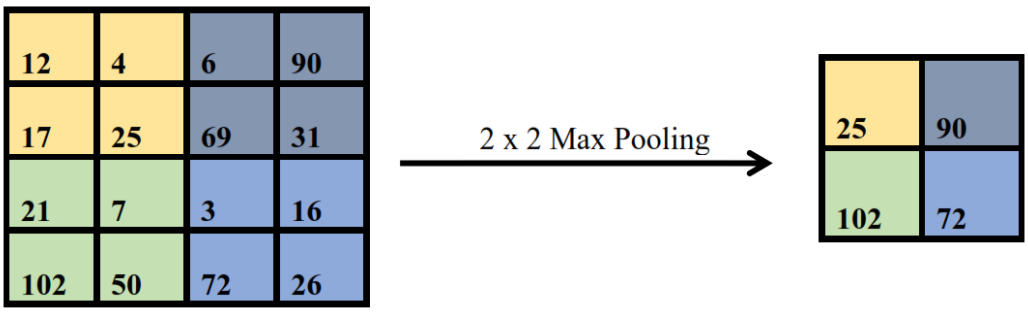


Figure 4. A picture depicting Max-pooling

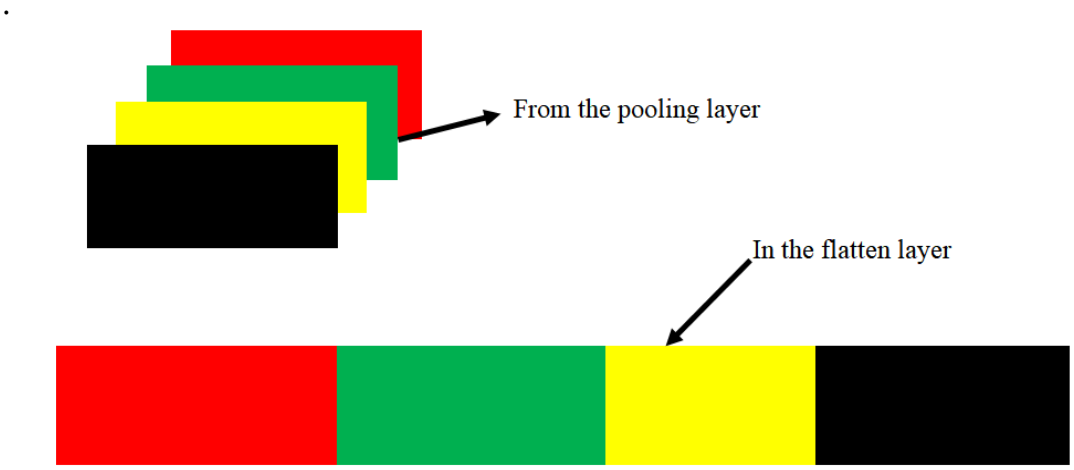


Figure 5. A picture showing the result of the flattened layer

The fully connected layer

The fully connected layer processes neurons from the previous layer [13]. The neurons in a fully connected layer and those from the previous layer are completely connected [17]. A fully connected network does the job of image classification at the end of the convolutional neural network [12]. A representation of the architectural framework of a convolutional neural network is depicted in Figure 6.

Activation Functions

The activation function is an important aspect of the deep neural network. At this stage, non-linear activation (non-linearity) is employed to ensure complex feature maps are well learnt during the course of training. If the non-linear activation function is left out, the fully connected layer will perform a dot product and an addition operation, which are linear [9]. This implies that linear functions do not have the required capacity to learn complex representations of regions in input data. The most commonly used non-linear activation in deep learning is the ReLu (Rectified Linear Unit) [9], which was applied in this research. ReLu is simplified in terms of operation; it is quite useful in the removal of the gradient problem [18]. It is expressed as **f(x) = max (0, x)**, which means that for any value below 0, ReLu assigns the value '0' to it and it assigns the actual value of any number above 0 as 'x' [18]. Softmax activation was used for the output layer because, in multiclass classification problems, it returns the image of the target class by assigning the highest value to it [19].

CNN vs ANN

CNN was preferred over Artificial Neural Network (ANN) in this research because, in an artificial neural network, all the neurons in a layer are fully connected to the neurons in the adjacent layer, and this could impact adversely on the resultant output, especially when the image size is large. The data set used in this research had an image size of [176 x 176 x 1]. In an artificial neural network, the image would be flattened into a vector of [176 x 176 x 1], which will result in 30,976 rows being used as the weight for a single image in a layer. This is quite large and could cause over-fitting, which is not fit for image processing [20].

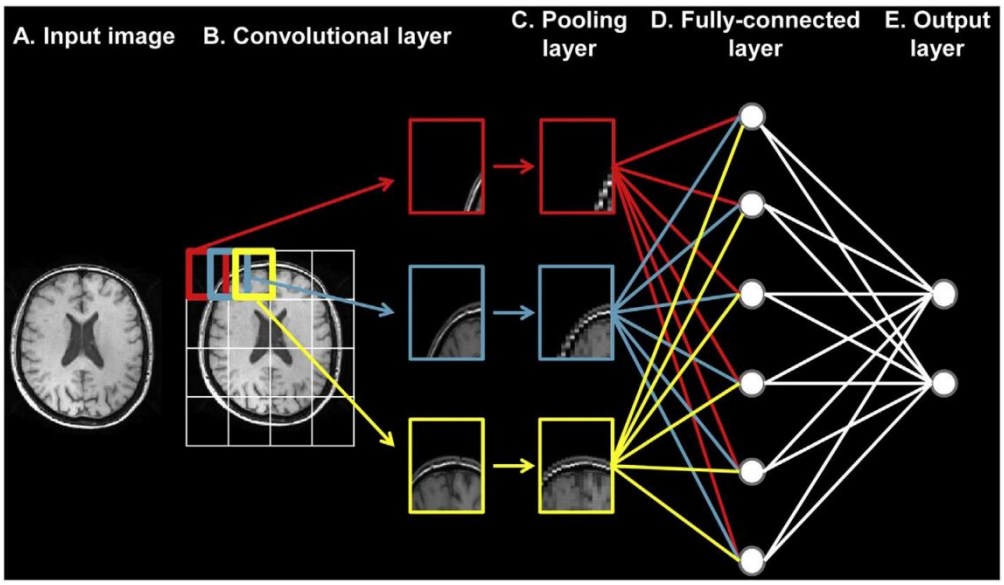


Figure 6. A typical Architecture of a convolutional neural network [7]

This is because an artificial neural network takes the full image feature to the next layer, unlike a convolutional neural network, which takes the spatial representation of the image. As a result of this operation, huge computational resources will be required to process and execute image-related problems using artificial neural networks [20]. Using a convolutional neural network in the same scenario is quite different because, in a convolutional neural network, all neurons are connected to each other at the last layer of the network. In convolutional neural networks, filters are used to learn spatial representations and generate a feature map. These feature maps are joined at the last layer to produce the output [20]. Considering the same scenario using a convolutional neural network, an image size of [176 x 176 x 1] and applying a filter of size [3 x 3 x 1] gives 9. This means that each neuron in a convolutional layer will be connected to only nine spatial representations of the image in that layer, which will eventually reduce computational cost [20].

Model Definition

The model comprised the VGG16, Flatten and Dense layers. The activation function, ReLu was used at the input layer to ensure nonlinearity existed. ReLu was chosen because, in a classification problem like this, the actual value of the output must belong to exactly one class. ReLu assigns values between 0 and >0, for which any value >0=x. Filter sizes of 512 and 1024 were used. A flatten layer was introduced before the dense layers to ensure the tensors are flattened to a one-dimensional vector before been fed into the dense layer. Dropout with a percentage of (0.1) was introduced to mitigate over-fitting. Softmax activation was used for the output layer because in multiclass classification problems, it returns the image of the target class by assigning its highest value [19]. Adam optimizer was used for this research since it combines the approaches used in stochastic gradient descent (SGD) and that of root mean square prop (RMSProp). It offers a better choice for optimization. A learning rate of 0.001 was used. In order to compile the newly built model, categorical cross entropy was used as the loss function because of the multiclass classification. It enabled the optimizer to adjust the weights in order to minimize the loss score. Accuracy was the metric adopted for the measurement of performance.

Dropout

Dropout, a form of regularization technique, is widely used to mitigate over-fitting [16]. The purpose of the dropout was to randomly remove (dropping out) some neurons during training, then trains with a less complex formed network. Thereafter, the dropped-out neurons are restored. This process continues until the model finds the best parameters, which were then used for better predictions [21]. Dropout was applied in the two dense layers and this greatly improved the performance of the model.

K-Fold Cross Validation

While training a model with a relatively small data set, it can be quite challenging with respect to the performance evaluation of the model in determining an ideal accuracy. K-Fold cross validation provides a solution to the above problem. K-fold was employed in Keras with the aid of the scikit-learn library [22]. This was done by dividing the dataset into k-number of segments in this case, it was 5-folds. One segment was used to validate the model while the remaining segments were used for training since more data was needed for training [22]. After each run, the validation score was taken and thereafter the averages of the scores obtained were taken as the validation score derived from the model [5].

RESULT AND DISCUSSION

Transfer Learning Technique

In this project, a transfer learning approach was used. It is a deep learning approach, where a part or whole of a trained model is used to train on the dataset which is to be used for the new model [23]. The sMRI dataset used for this research contained 178 samples. This was not ideal for training a deep neural network because it was small. Hence, a pre-trained VGG16 model trained in ImageNet was used to perform this operation. The VGG16 pre-trained model is embedded in Python; it has been prepackaged in Keras [9]. The VGG16 model won the ImageNet challenge in 2014 [24]. The model has 16 weight layers, including three fully connected layers [24].

Fine tuning the pre-trained model

In fine tuning the pre-trained model, weights and biases of the convolutional layers of the VGG16 pre-trained model were frozen to prevent it from updating the original weights during training. This was done to ensure that the representations (features) that the VGG16 model has learnt previously were prevented from being modified in the course of the training so it transferred to the dataset used in the new model (pre-trained model) [24]. This was done by setting the trainable attribute of the VGG16 model to **False**. The last three layers of the VGG16 model were replaced with three new dense layers. The original three fully connected layers of the VGG16 were popped out and replaced with a new one to suit the intended output of three classes. This ensured that the structure of the VGG16 model was retained.

Model Results

A pre-trained VGG16 model imported from Keras was used. The model was tested on two platforms. It was initially tested on CPU, thereafter, it was tested on GPU. On CPU, the model had an overall accuracy of 91.6%, while on GPU, the model had an upward performance of an overall accuracy of 94%.

Effect of weight balancing and dropout

Weight balancing and dropout formed a crucial part of the model. The model was seen to generalize well after the implementation of these two processes, which resulted in an improved performance of the model.

Effect of speed

In the course of finding out the best possible way to improve the speed of training, the Google Colab running on GPU was stumbled upon. It was discovered that the model trained faster on GPU, which gave enough room to observe the result, and necessary modifications were made as soon as possible. The resultant effect was that, with the improved speed, the model's performance also improved. This means that the rate of acceleration has effect on the performance of the model.

CONCLUSION

This paper proposed a machine learning technique with the use of a convolutional neural network to implement an image classification model. The model was to classify images of Alzheimer's disease from healthy control. The model used the transfer learning technique, where a VGG16 pre-trained model was employed to generate an output of multiclass classification of 3 classes. The sMRI dataset of 178 images used for the research was quite small and constituted a challenge in the research which was overcome subsequently. The

reason was that medical data were often difficult to come by as they involved samples from humans and as such requires full consent of the individuals involved. K-fold cross validation was employed to ensure the model had apparently similar amounts of data for training in each fold and for validating the model's performance. Class weight ensured the classes had proportional weights during training. The k-fold cross validation proved to be effective with a small amount of dataset available for the project. After training and fine tuning, the model had an overall accuracy of 94%. Further future studies and improvements to this work can be more attractive if there is enough data to be used in the training of the convolutional neural network. This is to ensure a more realistic and robust performance of the model, more so in a multiclass classification problem. In addition, the improvement could include a robust web-based platform that will enable users to enter a set of data (images), and the application outputs the result of the classification.

REFERENCES

[1] Alzheimer's Society, "Diagnosing Alzheimer's disease" [Online]. Available: https://www.alzheimers.org.uk/about-dementia/types-dementia/diagnosing-alzheimersdisease#content-start [Accessed: 24 May 2019]

[2] T. D. Bird, "Alzheimer Disease Overview. 1998 Oct 23 [Updated 2015 Sep 24]," GeneReviews®[Internet]. Seattle (WA): University of Washington, Seattle, 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBk1161/#alzheimer_escape_l_clinical_characteri [Accessed: 24 May 2019]

[3] M. C. Carrillo and H. Hampel, Alzheimer's Disease-Modernizing Concept, Biological Diagnosis and Therapy, *Advances in Biological Psychiatry*, 2012, doi: 10.1159/isbn.978-3-8055-9803-3

[4] Mayoclinic, "Alzheimer's disease" [Online]. Available: https://www.mayoclinic.org/diseases-conditions/alzheimers-diseases/diagnosis-treatment/drc-20350453 [Accessed: 17 July 2019]

[5] Expert System, "What is Machine Learning". [Online]. Available: https://www.expertsystem.com/machine-learning-definition [Accessed: 25 May 2019]

[6] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, ID: 7553, pp. 436, 2015, doi: 0.1038/nautre14539

[7] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, "Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks," *NeuroImage: Clinical*, vol. 21, 2019, doi: 10.1016/j.nicl.2018.101645

[8] K. R. Kruthika, H. D. Maheshappa and Alzheimer's Disease Neuroimaging Initiative, "Multistage classifier-based approach for Alzheimer's disease prediction and retrieval," *Informatics in Medicine Unlocked*, vol. 14, pp. 34-42, 2019, doi: 10.1016/j.imu.2018.12.003

[9] F. Chollet, *Deep Learning with Python*, New York: Manning Publications Co., 2018.

[10] G. Seif, "Handling Imbalanced datasets in Deep Learning", 19 November 2018. [Online]. Available: https://towardsdatascience.com/handling-imbalanced-datasets-in-deep-learningf48407a0e758 [Accessed: 17 July 2019].

[11] U. R. Acharya, L.O. Shu, Y. Hagiwara, J.H. Tan, H. Adeli, D.P. Subha, "Automated EEGbased screening of depression using deep convolutional neural network," *Computer Methods Programs Biomedicine*, vol. 161, pp. 103-113, 2018, doi: 10.1016/j.cmpb.2018.04.012

[12] M. A. Hossain and M. S. A. Sajib, "Classification of Image using Convolutional Neural Network (CNN)," *Global Journal of Computer Science and Technology*, vol. 19, no. 2, pp.13-14, 2019.

[13] F. Sultana, A. Sufian and P. Dutta, "Advancements in Image Classification using Convolutional Neural Network," *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, Kolkata, India, 2018, pp. 122-129, doi: 10.1109/ICRCICN.2018.8718718.

[14] D. Jaswal, S. Vishvanathan, K. P. Soman, "Image classification using convolutional neural network," International journal of scientific & engineering research," *International Journal of Advancements in Research & Technology*, vol. 5, no. 6, 2014, doi: 10.14299/ijser.2014.06.002

[15] M. Cavaioni, "Deep Learning series: Convolutional Neural Networks" 24 February 2018. [Online]. Available: https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524 [Accessed: 03 July 2019]

[16] Max-Pooling [Online]. Available: https://computersciencewiki.org/index.php/Max-pooling_/_pooling [Accessed: 30 June 2019]
[17] J. Hong S. Wang, H. Cheng, & J. Liu, "Classification of cerebral microbleeds based on fully-optimized convolutional neural network," *Multimedia Tools and Applications*, vol. 79, pp. 1-19, 2020, doi: 0.1007/s11042-018-6862-z
[18] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Computer Science*, vol. 131, pp. 977-984, 2018, doi: 10.1016/j.procs.2018.04.239
[19] C.E. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *2nd International Conference on Computational Sciences and Technology, (INCCST) 2020*, Jamshoro, Sindh Pakistan, 2020.
[20] I. Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, Chennai, India, 2017, pp. 1-6, doi: 10.1109/ICSCN.2017.8085675
[21] X. Ying, "An overview of overfitting and its solutions," in *Journal of Physics: Conference Series*, vol. 1168, ID: 022022, 2019, doi: 10.1088/1742-6596/1168/2/022022
[22] K. Hewa, "K-Fold cross validation", 16 December 2018. [Online] Available: https://medium.com/datadriveninvestor/k-fold-cross-validation-6b8518070833 [Accessed: 8 July 2019]
[23] J. Brownlee "A gentle introduction to transfer learning for deep learning" Machine Learning Mastery, 2017. [Online] Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning [Accessed: 25 June 2019]
[24] Kaggle, "VGG-16 pre-trained model for keras". [Online]. Available: https://www.kaggle.com/keras/vgg16 [Accessed: 25 July 2019]