Original article

Journal of High School Science

Convolutional autoencoders combined with kernel density estimation for data efficient and accurate disease detection in MRI scans of the brain

Aggarwal R

Submitted: June 13, 2023, Revised: version 1, August 8, 2023, version 2, October 11, 2023
Accepted: October 11, 2023

Abstract

Up to 5000 deaths occur each year in the United States from errors in diagnosis of neurological diseases. As most doctors often only check for certain common diseases, patients can go undiagnosed for other rare ones. This machine learning study may improve patient outcomes and thus decrease the number of patients who go undiagnosed, by developing a Convolutional AutoEncoder which consists of an Encoder and Decoder. The Encoder compressed the input of a Magnetic Resonance Imaging (MRI) scan into a lower dimension latent space. The Decoder reconstructed this lower dimensional representation back to the original image. The Encoder and Decoder were implemented in a symmetric multi-layered network along with MaxPooling and UpSampling to capture important features of the images. Trained on only healthy MRI scans of the brain, the model was able to detect the presence of any disease in the brain through a Kernel Density Estimation (KDE) score. KDE analyzed the output of the Encoder through a probability density function of each pixel. Using thresholds based on KDE score, this study achieved a 0 false negative, 100% accurate result for detecting unhealthy brains in MRI scans. These scans included diseases such as Alzheimer's, Pituitary tumors, Glioma tumors, and Meningioma tumors, all correctly classified as unhealthy even though the model was trained only on non-specific healthy images. Achieving a reasonable 90% accuracy on healthy images with 10% false positives, this AutoEncoder-KDE approach limited the need for large datasets. This approach may be applied to any medical image and decrease the number of undiagnosed patients.

Keywords

Machine Learning, AutoEncoder, Disease detection, Kernel Density Estimation, MRI scan, Neurological disease, Undiagnosed, Bottleneck, Brain tumor, Diagnosis error

---

Raghav Aggarwal, Westwood High School, 12400 Mellow Meadow, Austin, Texas 78750, USA.
raghavaggarwal926123@gmail.com

Introduction

Diseases going undiagnosed is a significant problem in the United States and upwards to 80,000 people each year are affected because of it, 6.6% of which were related to neurological diseases (1). On one hand common diseases such as cancers go undetected, and on the other hand a significant part of this number is because of a lack of awareness and data on rare diseases. This lack of awareness led to morbidity and mortality. Despite the reality of not having enough data on rare diseases, machine learning solutions have incorporated large amounts of data. Even for binary classification between, for example, Lung CT scans with Covid-19 and scans that are healthy, standard machine learning solutions required data on both classes (2). Furthermore, multiclass classification of diseases were only limited to the different diseases the models were trained on (3). This led to a high number of false results; images with a disease the model was not trained on were misclassified limiting the application of the model to only certain diseases. Training a model for classifying 100 different diseases could occur, however the challenge is the same when the model meets a disease it was never trained on. For rare diseases, a machine learning solution that required a lot of data would be impractical. Another implication of this research is to create a solution that will assist doctors and not replace them. A model that can accurately differentiate between a healthy scan and a scan with any disease without creating a high number of false positives; i.e. healthy images being diagnosed as unhealthy. In order to achieve this, this research utilized an AutoEncoder. An AutoEncoder consists of an Encoder and Decoder. The Encoder compresses input data into a lower dimension representation. The Decoder then reconstructs this lower dimension representation (4). An AutoEncoder is commonly used for disease detection by creating accurate low dimensional representations that then become the input for standard classifiers (5). This research utilized MRI scans of the brain in order to test an approach for identifying scans that have a disease that can extend to all parts of the body. This research utilized AutoEncoders strategically by training them only on healthy Magnetic Resonance Imaging (MRI) scans of the brain, allowing the system to detect brain MRI scans with any disease. As the AutoEncoder model is only trained with healthy images, it would learn to reconstruct healthy scans but not scans with diseases. Specifically, the area of the image that has the disease would contain an arrangement of pixels that the model would not have encountered. Hence, it would not be able to reconstruct that area as well, leading to a higher loss. However, a significant challenge with such anomaly detection frameworks is the high number of false positives they can generate (6). In order to solve this, this research also utilized Kernel Density Estimation. Kernel Density Estimation created a probability density function of the data which can then check against new data (7). By creating a probability density function of the healthy scans' pixels, unhealthy scans could be checked against the function for another threshold, thereby allowing for less false positives.

Methods

This work utilized 330 MRI scans of the brain across 4 disease classes: Alzheimer's, Glioma tumor, Meningioma tumor, and Pituitary tumor and 498 healthy scans (8-9). 395 of the healthy scans were used for training, while 103 were used for testing. This was to ensure that the model did not overfit on the healthy images.

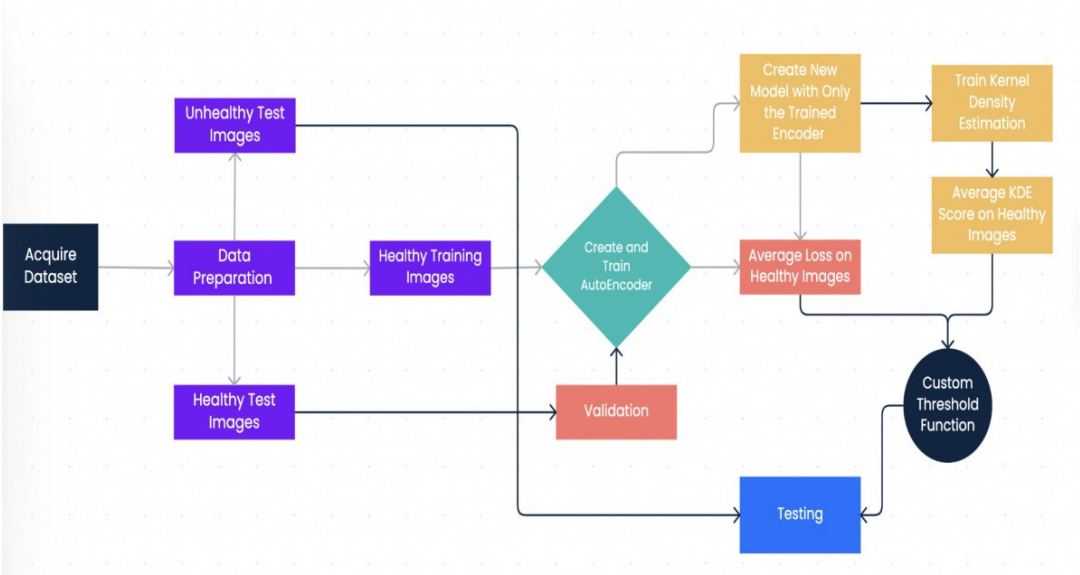

Figure 1. The pipeline of this research is shown. The steps included data preparation, model creation, threshold creation, and testing.

First, data preparation was needed in order to ascertain that all scans began at the same dimension. The data was also separated into different classes. The 3 classes were healthy scans used for training, healthy scans used for testing, and unhealthy scans used for testing.

AutoEncoder

Next, a Convolutional AutoEncoder was trained using the healthy training set. Unlike a standard Convolutional Neural Network (CNN) structure in which an image's important aspects are learned and stored, which are then transferred to a fully connected layer for classification; in the Convolutional AutoEncoder, the model consisted of an Encoder and Decoder both based of convolutional layers. The Encoder took an image, and reduced the dimensions of the image using convolutional layers. Similar to a CNN, important aspects of the image were learned. The lower dimension representation, also known as the bottleneck, was then transferred to a Decoder. The Decoder attempted to reconstruct the image based on only the bottleneck data.

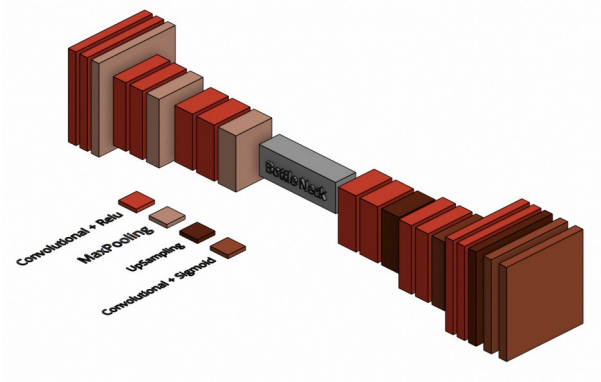

Figure 2. Model structure. The model architecture consisted of the Encoder which has 3 layers, and a Decoder which has 4 layers.

All the scans started at the size of 128x128 pixels. Then, after transferring into the first convolutional layer and a MaxPooling layer, the scans were altered to 64x64x32. MaxPooling helped to lower the dimension by choosing a pixel to represent a set of pixels. The 64x64x32, images were then transferred to another convolutional layer and MaxPooling, that converted them to size 32x32x16. Then a final transfer through another analogous layer decreased the scan size to 16x16x16. Subsequently, the Decoder, consisting of 3 convolutional layers combined with UpSampling, converted the image of size 16x16x16 to 32x32x16; then 64x64x32; and finally to 128x128 pixels.. A sigmoid activation function was used to ensure that all pixel values were between 0 and 1.

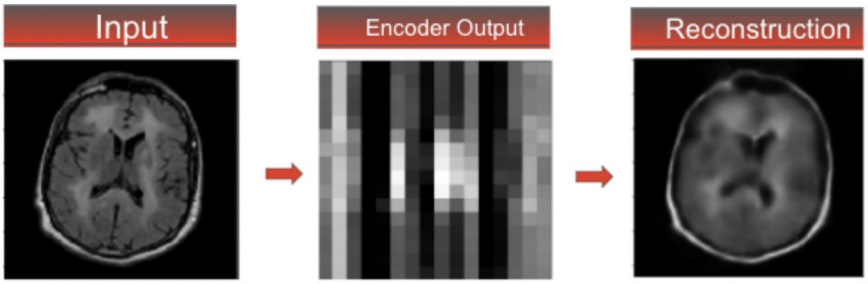

Figure 3. Input to Reconstruction. A healthy MRI scan of the brain was inputted, then the lower dimension representation was shown, and finally the reconstruction.

The reconstructed scans were then compared to the original scan using the Mean Squared Error (MSE) loss function. The difference between each pixel in the reconstructed image was subtracted from the original image. All these differences were averaged to get the Mean Squared Error loss.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

As the AutoEncoder was only trained on healthy images, the idea was that when a 'diseased' scan was transferred through, its MSE loss would be significantly greater than that from a healthy image. To create a threshold for this classification, the average loss and standard deviation of the loss on the trained healthy images were first calculated.

Kernel Density Estimation (KDE)

To address the problem of the high number of false positives typically seen in anomaly detection problems, Kernel Density Estimation was used as a second threshold for detecting scans with diseases.

The difference between the AutoEncoder and the Kernel Density Estimation was the function through which the inputs of the image were passed through. On one hand the AutoEncoder attempted to reconstruct the input numbers to a new output which was as similar to the input as possible (after first reducing the dimension), while on the other hand KDE "plotted" all the input pixel values on a probability density function to see how likely it was to be present in a set of data. Both methods were able to account for continuous, as well as discrete values.

Kernel Density Estimation was applied to the output of the trained Encoder part of the AutoEncoder to enhance the performance of the probability function created.

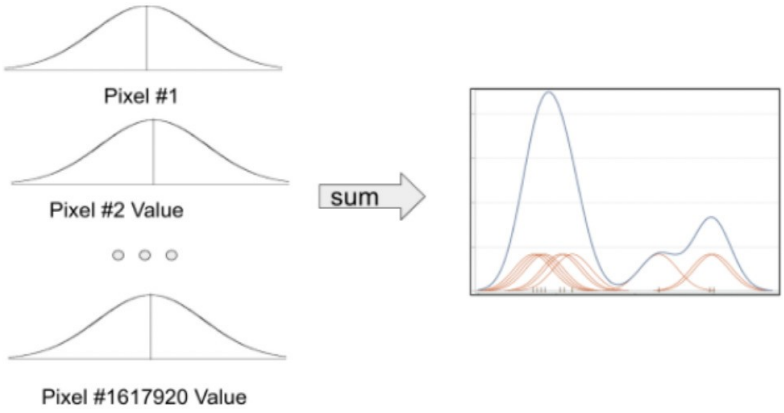

Figure 4. KDE representation. The diagram shows a graph representing the process of Kernel Density Estimation. A distribution function is created for each pixel. The Encoder output is 16x16x16 and there were 395 images used for training, or a total of 1617920 pixel values. All the distribution functions were added together to create a probability density function of all the healthy images pixel data.

Kernel Density Estimation was applied to the output of the trained Encoder for two reasons. First, the pixels that did not contain the brain would be part of all scans, healthy and unhealthy. Hence, by using a lower dimension representation, the number of such pixels was reduced. Furthermore, since the Encoder was trained, it recognized and captured the important features of the scan; if it did not succeed in capturing the important features, then it would not be able to reconstruct scans well and not overfit. Since the Encoder output likely contained important features, it was likely to contain arrangements of pixels which would not be found in scans of unhealthy brains.

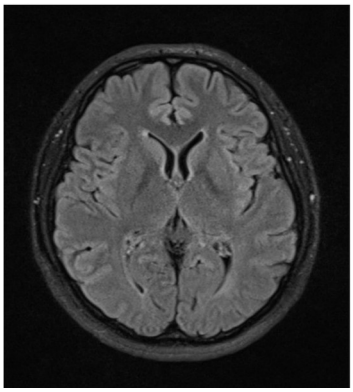

Figure 5. MRI Scan of healthy Brain.

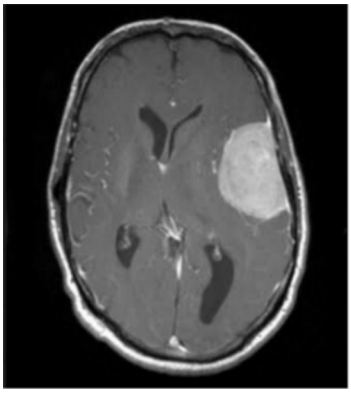

Figure 6. MRI Scan of unhealthy Brain with a meningioma tumor

By using KDE, access was obtained to a probability density function that was based on the pixels of the lower dimension representation of healthy images. Next, using this function the KDE score was calculated. Each pixel in the scan was measured for its probability to occur in that function. For example, a scan with a Meningioma Tumor was transferred to the AutoEncoder. The output of the Encoder was stored and had its pixels compared against those of the probability function. In other words, for each of 4096 pixels of the bottleneck representation (16x16x16) the probability of each pixel being in that distribution was checked. All the probabilities were added to get a score. Hence, by applying this technique on all the healthy training scans, another threshold was created – by finding the average KDE score on the healthy scans and standard deviation. If a scan were to be found outside this threshold it would most likely be a scan with a disease. Therefore, using both the loss threshold and KDE threshold, the testing data which consisted of scans from 4 different disease classes as well as healthy scans was tested.

Results and Discussion

In order for the thresholds to accurately differentiate between diseased and healthy brain scans, it was important to find a right balance, where the AutoEncoder reconstructed the healthy images, but not the unhealthy images. Furthermore, the chosen thresholds

needed to ensure that false positives were minimized and that all unhealthy scans were diagnosed correctly.

Trial	Epochs	Average Loss	Loss STDEV	Average KDE Score	KDE STDEV	Loss Threshold	KDE Threshold
14	500	0.008437	0.004589	2822.615	0.5102	NA	2800
13	500	0.008437	0.004589	2822.615	0.5102	0.0085	2800
12	500	0.008437	0.004589	2822.615	0.5102	0.0085	2820
11	300	0.008794	0.004399	2822.513	0.37	NA	2800
10	300	0.008794	0.004399	2822.513	0.37	0.009	2800
9	300	0.008794	0.004399	2822.513	0.37	0.009	2820
8	150	0.010492	0.006149	2822.6505	0.5344	0.011	2800
7	150	0.010492	0.006149	2822.6505	0.5344	0.011	2820
6	50	0.014594	0.007462	2822.5774	0.4305	0.015	NA
5	50	0.014594	0.007462	2822.5774	0.4305	NA	2820
4	50	0.014594	0.007462	2822.5774	0.4305	0.015	2820
3	50	0.017928	0.008794	2822.6048	0.4743	NA	2820
2	20	0.017928	0.008794	2822.6048	0.4743	0.02	2820
1	20	0.017928	0.008794	2822.6048	0.4743	0.0179	2820

Figure 7. Thresholds. In order to ensure the best results, different epochs of training allowing for different average losses on the healthy train images and different average KDE scores were tested, which in turn allowed for different thresholds as shown.

Trying to find the best threshold also included testing whether a loss only threshold or KDE score only threshold generated better results than different combinations of the two thresholds.

Trial	Healthy Brain	Glioma Tumor	Meningioma Tumor	Pituitary Tumor	Alzheimer's	Accuracy
14	90%	100%	100%	100%	100%	97.46%
13	82%	100%	100%	100%	100%	95.61%
12	22%	100%	100%	100%	100%	81.52%
11	90%	100%	100%	100%	100%	97.46%
10	81%	100%	100%	100%	100%	95.38%
9	5%	100%	100%	100%	100%	77.37%
8	86%	100%	100%	100%	100%	96.70%
7	25%	100%	100%	100%	100%	82%
6	85%	38%	54%	34%	6%	49.19%
5	19%	100%	100%	100%	100%	80.60%
4	19%	100%	100%	100%	100%	80.60%
3	20%	100%	100%	100%	100%	80.80%
2	20%	100%	100%	100%	100%	80.80%
1	19%	100%	100%	100%	100%	80.60%

Figure 8. Results for each trial. The table above shows the different accuracy of each individual class and the overall accuracy for each trial. A different threshold was used for each trial. This included thresholds that only used the AutoEncoder with MSE loss, only the KDE score, and also thresholds that used both. Specific thresholds are shown in Figure 7.

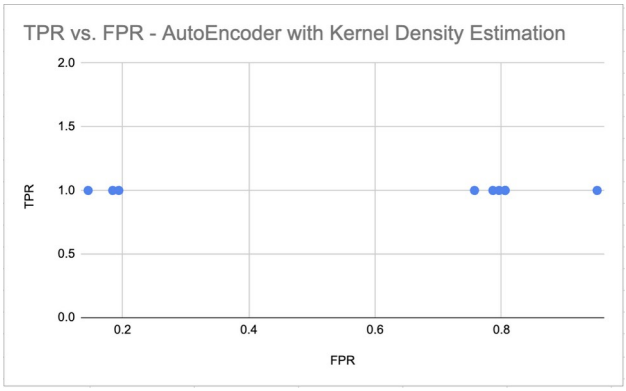

Figure 9. ROC Curve for AutoEncoder with Kernel Density Estimation trials. The graph shows the ROC curve for the thresholds which used both the AutoEncoder and Kernel Density Estimation. The true positive rate stays consistent throughout all trials, while the false positive rate decreases. The horizontal ROC shows that this AutoEncoder and KDE approach does not have to tradeoff between sensitivity and false positives, rather, the tradeoff is only in becoming more conservative in what the algorithm deems to be a disease. The AUC is .8058 which indicates that this approach arrives at accurate results a signficantly greater number of times than would be possible by chance alone.

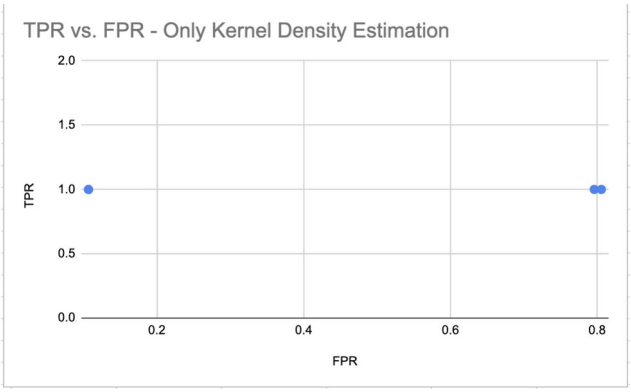

Figure 10. ROC Curve for only Kernel Density Estimation trials**.** The graph shows the ROC curve for the thresholds which used only Kernel Density Estimation. Again the true positive rate stays consistent throughout all trials, while the false positive rate decreases. The AUC is .7961, which again shows the KDE arrives at accurate results a signficantly greater number of times than would be possible by chance alone. The AUC is lower than that of the AutoEncoder and KDE system, probably because fewer trials were performed.

A ROC curve was not calculated for AutoEncoder-only trails because as seen in the one trial which only used the AutoEncoder; the sensitivity or true positive rate dropped significantly. The AutoEncoder alone did not accurately differentiate between images of the healthy and unhealthy brains.

Observations

From as low as 20 Epochs the model detected scans with diseases with 0% false negatives. However, at low epochs, the model was only slightly better at reconstructing healthy images than unhealthy images; a loss-only threshold led to large amounts of false positives. Some healthy images had a high loss, however at low epochs of training a KDE-score only threshold still led to a high amount of false positives.

On the other hand at high epochs of training the reconstruction loss of healthy images was measurably lower than unhealthy images. This was seen through a reduction in false positives when using a KDE and loss threshold after at least 300 epochs of training. However, many healthy images had a loss much higher than the average reconstruction loss of healthy images. But there was a significant difference between the distribution (KDE score) of healthy images and unhealthy images. A KDE-only threshold thus gave the best results. The probability density function generated at high epochs of training of the AutoEncoder was best able to differentiate between healthy and unhealthy images.

Results

The best results were achieved with a KDE-only threshold at 500 epochs of training. This resulted in an overall accuracy of 97.46% and 0 false negatives across 4 disease classes: Glioma tumor, Meningioma tumor, Pituitary tumor, and Alzheimers. However, if a KDE that was not based on the Encoder outputs was trained, i.e. it was trained on all pixels in the regular dimension representation, and which used a KDE only threshold, then an accuracy of only 77.14% was achieved. Hence, the best approach was when a KDE-only threshold was used, with a KDE that was trained on the outputs of a trained Encoder. It was necessary to train the AutoEncoder for optimal results.

To better understand the results, the recall, precision, and F1 Score were also calculated.

$$\text{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives} = \frac{330}{330+0} = 100\,\%$$

$$\text{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives} = \frac{330}{330+11} = 96.8\,\%$$

$$\text{F1 Score} = \frac{Precision * Recall}{Precision + Recall} \times 2 = \frac{.9677 \times 1}{.9677 + 1} \times 2 = 98.4\,\%$$

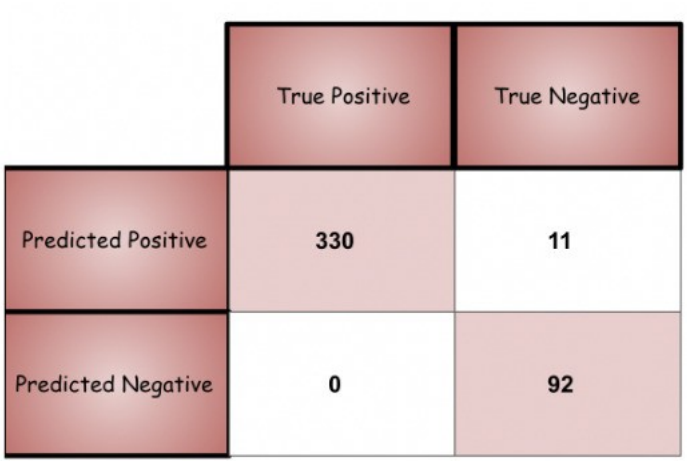

Figure 11. Confusion Matrix. A confusion matrix representing the final results. True Positive refers to a scan that has a disease and True Negative refers to a healthy scan. Out of 330 scans that had a disease, all 330 were predicted as having a disease. Out of 103 scans that were healthy, 92 were classified as healthy while 11 healthy scans were classified as having a disease.

Discussion

The KDE only threshold, in which the Kernel Density Estimation was trained on the outputs of the Encoder, which, in turn was trained for 500 epochs, was able to reach a 100% recall rate without producing a significant amount of false positives. The F1 score was > 98%. The F1 score gives more importance to eliminating false negatives than false positives (diseases going undiagnosed is penalized more). The model was hence found to be high performing at eliminating the problem of diseases going undiagnosed, but at the same time not causing many false positives.

This approach outlined in this paper can potentially be transferred to other medical images or anomaly detection problems. The only difference would be what the AutoEncoder and Kernel Density Estimation are trained on. For instance, if trying to identify diseases in the lung, the AutoEncoder would be trained on healthy lung scans.

Overall, this solution of a Convolutional AutoEncoder combined with Kernel Density Estimation to some extent exceeded the performance of previous machine learning solutions that generated many false positives and doctors that produced many false negatives (10). It did this while only being trained on healthy images; an efficient and effective solution.

Limitations

Though this research assisted in disease identification it can only assist with identifying that a scan has a disease and not the specific

diagnosis. In order for this approach to be able to identify specific diseases, the methodology would need to be applied such that the system is trained on the disease. The algorithm can then classify that scan as having the attributes of that specific disease without the need to train on healthy images. The cost to implement this research in practice may be impractical in light of the cost of MRI scans. MRI scans of the brain can cost up to $8,400, abdominal $7,600, breast $10,300, and the chest $7,900 (11). Furthermore, preventive diagnosis screening for brain tumors, is not covered by health insurance like Medicare. (12) With the average cost of an MRI scan being $1,325 in the USA (13), performing for instance, an MRI scan on a number like 80,000 cases would lead to a cost of over $100 million. This does not include the time and resource effect of conducting so many scans. However, conducting these MRI scans would come with the potential benefit of doctors being sure that a patient has the disease.

Another consideration is that MRI scans, which would produce the scan to input through the system expressed in this paper, are often conducted after preliminary testing through which the doctor is sure of some abnormality in the patient. In order for this research to perhaps be more applicable, disease localization must be incorporated. Through techniques such as seeing where the reconstruction is the worst, the specific location of the disease can be identified producing a greater benefit for doctors, assuming that geometry specific partical scans can can be conducted and will cost less.

In summary it is unlikely that insurance companies or other 3rd party payers would adopt this universal MRI scan, pre diagnosis approach to perform analysis using the outlined method in this paper. Rather litigation costs of the misdiagnoses may cost less than universalizing MRI scans at entry point. In fact, neurological diseases such as Alzheimer's and Parkinson's can be diagnosed through paper cognitive assessment. In such cases MRI scans serve as confirmation of diagnosis, not as a method of diagnosis. As MRI scans serve often as confirmation, comparing MRI scans with other unhealthy scans of the disease at question will be more effective.

Conclusion

A convolutional autoencoder machine learning model was developed to differentiate between MRI brain scans of healthy persons and those with neurological diseases such as Alzheimer's disease, Pituitary tumors, Glioma tumors, and Meningioma tumors. The Encoder and Decoder comprising the convolutional autoencoder were implemented in a symmetric multi-layered network along with MaxPooling and UpSampling to capture important features of the images. Trained on only healthy MRI scans of the brain, the model was able to detect the presence of any disease in the brain through a Kernel Density Estimation (KDE) score. Using thresholds based on KDE score, this study achieved a 0 false negative, 100% accurate result for detecting unhealthy brains in MRI scans. This approach may be applied to any medical image to decrease the number of undiagnosed patients and to reduce errors in diagnosis.

References

1. “Diagnosis Journal” *Society to Improve Diagnosis in Medicine*, https://www.improvediagnosis.org/diagnosis-journal/

2. Zhang, X., Han, L., Sobeih T., Han, L., Dempsey, N., Lechareas, S. “CXR-Net: A Multitask Deep Learning Network for Explainable and Accurate Diagnosis of COVID-19 Pneumonia from Chest X-ray Images.” *IEEE J. Biomed. Health Inform.* 27(2), 980-991, (2023), https://doi.org/10.1109/jbhi.2022.3220813

3. Irmak, E. “Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework.” *Iranian J. Sci. Tech., Trans. Electrical Eng.,* 45. 1015-1036, (2021), https://doi.org/10.1007/s40998-021-00426-9

4. Jordan, Jeremy. “Introduction to autoencoders.” *Jeremy Jordan*, 19 March 2018, https://www.jeremyjordan.me/autoencoders/

5. Song, Q., Zhao L., Luo, X., Dou, X. “Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images.” *J. Healthc Eng.* 2017, 8314740, (2017), https://doi.org/10.1155%2F2017%2F8314740

6. Raza, K., Singh, N., K.. “A Tour of Unsupervised Deep Learning for Medical Image Analysis.” arXiv:1812.07715, https://doi.org/10.48550/arXiv.1812.07715

7. *Lecture 6: Density Estimation: Histogram and Kernel Density Estimator 6.1 Histogram*, http://faculty.washington.edu/yenchic/18W_425/Lec6_hist_KDE.pdf

8. Dubey, S. “Alzheimer's Dataset (4 class of Images).” Kaggle, https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images

9. Sartaj. “Brain Tumor Classification (MRI).” *Kaggle*, https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

10. Cook, A.,J., Elmore, J.,G., Zhu. W., Jackson, S.,L., Carney, P.,A., Flowers, C., et al. Mammographic interpretation: radiologists' ability to accurately estimate their performance and compare it with that of their peers. *AJR Am J. Roentgenol.,* 199(3), 695-702, (2012), https://doi.org/10.2214/ajr.11.7402

11. Woodard, D., Jacobson, A.. “How Much Does an MRI Cost (With/Without Health Insurance)?” GoodRx, 7 July 2022, https://www.goodrx.com/health-topic/diagnostics/how-much-does-an-mri-cost

12. “preventive services - Your Medicare Coverage.” Medicare, https://www.medicare.gov/coverage/preventive-screening-services

13. “How Much Does an MRI Cost Without Insurance in 2023?” Mira, 24 June 2023, https://www.talktomira.com/post/how-much-does-an-mri-cost-without-insurance-in-2021