E-ISSN: 2583-9667
Indexed Journal
Peer Reviewed Journal
https://multiresearchjournal.theviews.in

Received: 15-06-2023
Accepted: 22-07-2023

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH IN MULTIDISCIPLINARY

Volume 1; Issue 1; 2023; Page No. 483-492

A brief analysis of thresholding-based segmentation through deep neural networks using optimal Kapur's model for the detection and classification of brain tumor

[1]Mrutyunjaya and [2]Dr. Manish Saxena

[1]Research Scholar, Department of Computer Science, Himalayan University, Arunachal Pradesh, India
[2]Assistant Professor, Department of Computer Science, Himalayan University, Arunachal Pradesh, India

DOI: https://doi.org/10.5281/zenodo.12795546

Corresponding Author: Mrutyunjaya

Abstract

Technology has significantly impacted the medical sector, helping to prevent, treat, and monitor a wide range of patient conditions. The medical field has seen several improvements and increased sophistication since the advent of machine learning, deep learning, and artificial intelligence based algorithms and models. Applications for computer-aided diagnosis have automated thousands of medical diseases, and life-saving interventions will inevitably follow suit. The medical community places a great deal of emphasis on the identification and categorization of tumours, which it claims are the main causes of death in all age categories. The manual labor required to gather, record, categorize, and then arrange for an accurate diagnosis is laborious and time-consuming, which increases the possibility of mistakes. With the need of early identification and prompt treatment for patients with various tumor types, the field has welcomed sophisticated, automated solutions that harness the power of artificial intelligence and machine learning algorithms. Modern medical imaging technologies have made brain tumor identification and categorization less complicated, but radiologists and other related medical specialists still have a lot of work ahead of them. Maximum accuracy is used to detection and classification, and special attention is paid to how medicine affects progression or regression. The accuracy of manual detection methods varies with years of expertise and qualifications; hence, there is a growing need for computer-aided diagnosis procedures, which has created new avenues for research and development.

Keywords: Modern medical imaging technologies, brain tumors

Introduction

The OS-DNN model, which stands for Optimal Kapur's thresholding based segmentation with DNN, is a novel approach to BT detection and classification that is presented in this paper. Preprocessing, segmentation, feature extraction, feature reduction, and classification are the five main phases of the suggested OS-DNN model's operation. Three methods are used to preprocess MRI brain images: skull stripping, contrast enhancement based on Contrast Limited Adaptive Histogram Equalization (CLAHE) based on bilateral filtering (BF) based on noise removal, and one other. Next, the best segmentation procedure based on Kapur's thresholding is carried out, and the threshold value is optimized using the Grey Wolf Optimization (GWO) approach. After that, kernel principle component analysis (KPCA) is used for feature reduction and discrete wavelet transforms (DWT) are used for feature extraction. Ultimately, the MRI brain images are classified as normal or pathological using the DNN model. The OS-DNN model is experimented with on a benchmark Kaggle dataset, and performance analysis is used to look at the outcomes. As demonstrated by the simulation results, which have a maximum sensitivity of 97.94%, specificity of 98.08%, and accuracy of 98.02%, the OS-DNN model is superior to the other models.

The proposed OS-DNN model

Figure-1 shows the general process of the OS-DNN approach in action. The proposed OS-DNN model is shown to go through pre-processing, segmentation based on GWO-KT, feature extraction relying on DWT, feature reduction based on KPCA, and classification utilizing DNN model. The sections that follow provide an explanation of these procedures.

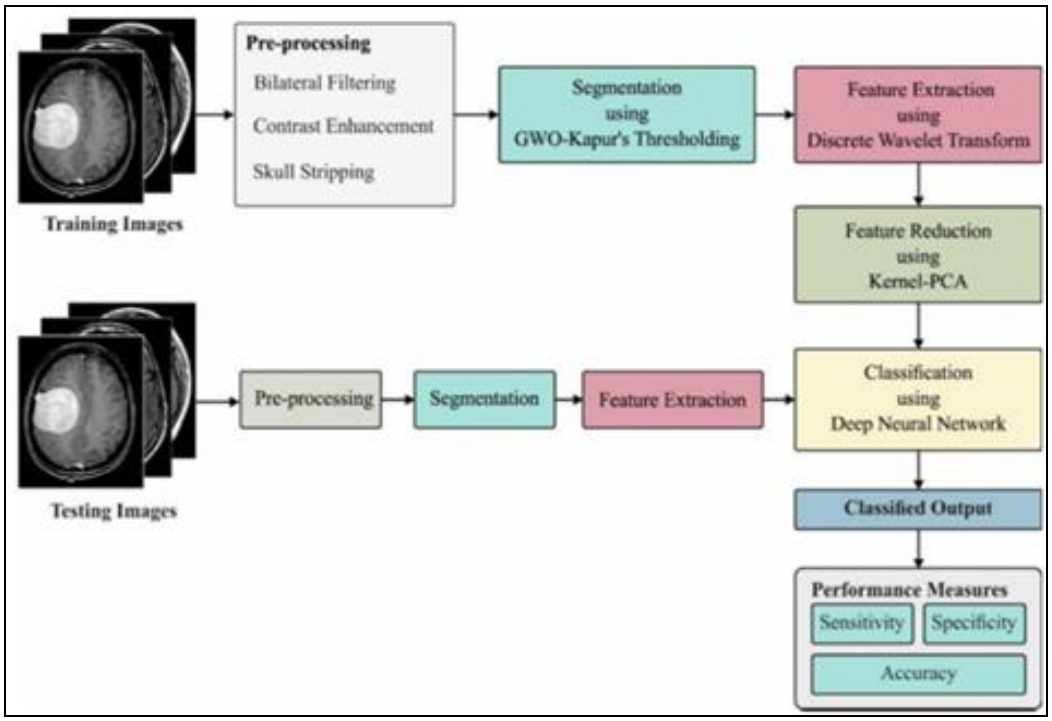

Fig 1: Working principle of OS-DNN model

Preprocessing

The input image is first preprocessed in three steps: skull stripping, CLAHE-based contrast enhancement, and BF-based noise removal. The discussion of these subprocesses follows.

Bilateral Filtering (BF)

The BF image smoothing filtering approach is non-linear, noise-removing, edge-conserving, and noise-conserving. It is used to apply a weighted average of intensity values from closer pixel values to replace the intensities of all pixels. The Gaussian distribution affects the weights. The BF approach can be basically expressed as follows.

$$I^{filtered}(x) = \frac{1}{W_p} \sum_{x_u \epsilon \Omega} I(x_u) f_r(\| I(x_u) - I(x) \|) g_s(\| x_u - x \|)$$

Where W$p$indicates the normalization term, which can be represented by

$$W_p = \sum_{x_i \epsilon \Omega} f_r(\| I(x_i) - I(x) \|) g_s(\|x_i - x\|))$$

Where the filtered image is denoted by $Ifiltered$, the actual input image is represented by I, the coordinate of the current pixel is x, fi is the window centering in x, xu ϵ fi is the other pixel, r is the range kernel to smoothen variation in intensity levels, and gs is the spatial (or domain) kernel to smoothen variability in coordinate points. The geographical proximity and intensity differences are given weights Wp. Let us assume that there is a pixel at coordinate (u, v) that needs to be denoised in the image using neighboring pixels. One of the nearby pixels is located at (k, l). Subsequently, the Gaussian kernels are applied to the range and spatial kernels. The weight assigned to pixel (k, l) for denoising pixel (u, v) can be expressed as

$$w(u, v, k, l) = \exp\left(-\frac{(u-k)^2 + (v-l)^2}{2\sigma_d^2} - \left(\frac{\| I \ (u, v) - I(k, l)\|^2}{2\sigma_r^2}\right)\right)$$

where the smoothing variables are σd and $\sigma \boldsymbol{U}$, and the implied pixel intensities are $I(u, v)$ and $I(k, l)$ respectively. Following the computation of the weights, the following normalizing procedure is followed.

$$I_D(u, v) = \frac{\sum_{k,l} I(k, l) w(u, v, k, l)}{\sum_{k,l} w(u, v, k, l)},$$

where ID is the denoised intensity of pixel (u, v).

Contrast enhancement using Clah model

This definition of Adaptive Histogram Equalization (AHE) is a computer image processing technique used to raise the contrast level of an image. By using an adaptive model, it differs from standard Histogram Equalization. It is applied to re-share lightness of image values and estimates different histograms in which it relates to distinct image parts. As such, it can be used to enhance edge definition and maximize local contrast in any given image.

Properties of AHE

This model's metric is thought to be the size of the neighborhood. It is made up of a particular length scale where contrast is limited at maximum scales and enhanced at tiny scales.

A pixel's output value from AHE is proportionate to the grade from neighboring pixels because of basic HE. It allows for efficient execution on high-end hardware by

comparing intermediate pixels with other pixels in the same neighborhood. Consolidating two pixels with the lowest value and adding one for each pixel with a comparable value yields the un-normalized result.
The histogram would be dynamically peaked when the image region had homogeneous pixel neighborhoods. Additionally, the conversion function converts the limited range of pixel values to the full range of the output image. It results in an extremely homogeneous image's smallest amount of noise being over-amplified, or AHE.
Because a normal AHE is primarily based on a histogram, it overamplifies the contrast in a closer, consistent image region. A version of AHE known as CLAHE reduces noise amplification and has limited contrast amplification.
Before calculating the Cumulative Distribution Function (PDF), CLAHE trims the histogram at a predetermined value to lessen the amplification. The transformation function has been used to lower the CDF's slope. After the histogram is normalized and the neighborhood region is taken into consideration, the value that is present in the histogram is clipped and given the designation "clip limit." It is advantageous and leaves the part of a histogram that exceeds the clip limit intact. All histogram bins are processed for the redistribution, though. The redistribution pushes some bins over the clip limit, producing an efficient clip limit that is maximum when compared to the precise value of a picture and its predefined extend.

Skull Stripping

It is the first step that must be completed before the brain MRI image segmentation procedure. The skull must be taken out of the MRI's backdrop area. The skull is separated using an image filtering technique, and the remaining picture portions are masked with pixels of the same intensity levels. The bone region of an MRI picture often has a higher threshold value than the normal and abnormal regions. As a result, the brain regions are divided using the image filtering approach based on a predetermined threshold value. Afterward, the skull is eliminated from the picture by using the solidity feature.

GWO-KT based segmentation

Using the entropy of the histogram as a guide, Kapur's image segmentation technique was mostly used in the early decades to separate grayscale photos. It finds the ideal T value to maximize the total entropy. Let T = [t1, t2,.., tk−1] be an image threshold vector. Next, the representation of Kapur's entropy would be

$$\mathrm{J}_{\max} = f_{kapur}(T) = \sum_{j=1}^{k} H_j^C \; for C\{1,2,3\}$$

Entropy is essentially calculated independently based on the t-value. The problem of multi-level thresholding can be illustrated as follows:

$$H_1^C = \sum_{j=1}^{t_1} \frac{Ph_j^C}{\omega_0^C} \ln\left(\frac{Ph_j^C}{\omega_0^C}\right)$$

$$H_2^C = \sum_{j=t_{1+1}}^{t_2} \frac{Ph_j^C}{\omega_1^C} \ln\left(\frac{Ph_j^C}{\omega_1^C}\right)$$

$$\vdots$$

$$H_k^C \sum_{j=t_{k+1}}^{L} \frac{Ph_j^C}{\omega_{k-1}^C} \ln\left(\frac{Ph_j^C}{\omega_{K\text{-}1}^C}\right)$$

The GWO approach has developed from the way that grey wolves hunt. These wolves live in packs of five to twelve and are thought to be elite predators. The wolves are divided into alpha (α), beta (β), delta (δ), and omega (ε) groups according to their preference for hunting. Dominant wolves are initially designated as the team's original dominators since they are able to make judgments based on factors like hunting, residence, and other factors. We call it the wolves of dominance. Alpha wolves set the rules, and other wolves abide by them. When there are no alpha wolves present, beta wolves step forward to make choices. It is granted the authority to look for solutions for alpha wolves and responds appropriately. Thirdly, the subordinate wolves, also referred to as subsequent level wolves, are the delta wolves. It falls among the categories of caretakers, hunters, scouts, elders, and sentinels.
Omega is finally the top dog and is used as a scapegoat. It uses the procedure that Alpha Wolves recommends. The omegas aid in internal problem solving and are ineffectual wolves. Wolf hunting can be divided into three phases: tracking and pursuing, pursuing and surrounding until the victim is attacked. Exploration and exploitation phases are used to operate GWO. The best solutions in the local search space are explored during the exploitation phase. While the prey is looking for better answers in the nearby search space, the grey wolves encircle and attack it. The wolves locate their prey and attack them when they are encircled. This method involves assuming the prey's position vectors and identifying the agents that move in accordance with the achieved optimal solution, which is shown as follows:

$$\vec{D} = \left|\vec{C}.\overrightarrow{Q_p}(n) - \vec{Q}(n)\right|$$

$$\vec{Q}(n+1) = \overrightarrow{Q_p}(n) - \vec{A}.\vec{D}$$

where prey position vectors are represented as $\overrightarrow{\bar{Q}\bar{\bar{p}}}(n)$, $\overrightarrow{Q}$ indicates the position vector, ‖ signifies absolute value, and n is the most recent iteration. demonstrates a step-by-step improvement. The vectors $\vec{A}$ and $\vec{C}$ can be written as follows:

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a}$$

$$\vec{C} = 2.\vec{r}$$

Grey wolves only contribute to β and δ during the hunting phase; α is the trigger. In the hunting phase, the optimum solution is not identified at first since it uses the most energy. The primary better candidate solution is represented by α, the upcoming optimal candidate solution is defined by beta, and the consequent better candidate solution is represented by delta. The hunting phase is demonstrated by the three solutions that are reached and extended to move lower ranked solutions:

$$\vec{D}_{\alpha} = |\vec{C}_1 * \vec{Q}_{\alpha} - \vec{Q}\ |$$

$$\vec{D}_{\beta} = |\vec{C}_2 * \vec{Q}_{\beta} - \vec{Q}\ |$$

$$\vec{D}_{\delta} = |\vec{C}_1 * \vec{Q}_{\delta} - \vec{Q}\ |$$

where the three coefficient vectors applied in altering distance vectors are $\vec{C}1$, $\vec{C}2$, and $\vec{D}$ð, and the altered distance vectors among the α, β, and γ places to alternate wolves are $\vec{D}\alpha$, $\vec{D}\alpha$, and $\vec{D}$ð. $\vec{Q}$ relates to the gray wolf's vector location (omega).

$$\vec{Q}_1 = \vec{Q}_{\alpha} - \vec{A}_1 * \vec{D}_{\alpha}$$

$$\vec{Q}_2 = \vec{Q}_{\beta} - \vec{A}_2 * \vec{D}_{\beta}$$

$$\vec{Q}_3 = \vec{Q}_{\delta} - \vec{A}_3 * \vec{D}_{\delta}$$

where $\vec{Q}$1 denotes a novel position vector obtained by applying the distance vector $\vec{D}$ð and the α position $\vec{Q}\alpha$, $\vec{Q}$2 indicates that a novel position vector is achieved under the application of the β position $\vec{Q}\beta$ and the distance vector $\vec{D}\beta$, $\vec{Q}$3 denotes the novel position vector obtained by applying delta position $\vec{Q}$ð along with the distance vector $\vec{D}$ð, and $\vec{A}$1, $\vec{A}$2, and $\vec{A}$3 are the three coefficient vectors assessed.

$$\vec{Q}\ \ (n+1) = \frac{\sum_{i=1}^{k} \vec{Q}_i}{n}$$

Where, β, and δ (k=3) are used to estimate the new position vector, which is shown by $\vec{Q}$ $(n + 1)$.

DWT based feature extraction

Generally speaking, DWT makes effective use of both dynamic scales and positions to deploy WT. Let (t) stand for a square-integrable function. The continuous WT of (t) with respect to the given wavelet $f(t)$ can be written as

$$W_{\psi}(a,b) = \int_{-\infty}^{\infty} u\,(t)\psi_{a,b}(t)dt$$

Where

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-a}{b}\right)$$

Using translation and dilation, where a indicates a dilation factor and b represents a translation parameter, the wavelet (t) is obtained from native wavelet (t). One important wavelet that is widely used in many different sectors and is relatively basic is the Harr wavelet.

Feature reduction using KPCA

Massive features result in a maximum processing time and increased storage needs. In addition, it is necessary to reduce the feature count and contributes to the "curse of dimensionality," which makes the classification process challenging. With the use of kernel models, PCA has been expanded into KPCA. The actual linear PCA work was conducted in regenerating kernel Hilbert space under kernel application. It is observed that, in the event that N points are not able to be classed generally, they are linearly classified in $d < N$ dimensions and significantly separated in a linear fashion in $d \geq N$ dimensions. The PCA component can be built for clustering. When the provided N points xi are mapped into N-dimensional space using

$$\Phi(\mathrm{u}_i)\ where\ \Phi: R^d \rightarrow R^N,$$

It is essentially produced as a hyper-plane that divides the points into arbitrary clusters. Since \ naturally creates the autonomous vectors at random, linear PCA cannot explicitly handle covariance in Eigen decomposition.

The random ¸ function that the KPCA chooses is non-trivial and cannot be explicitly approximated. When the real evaluation is not possible in data space, it activates the chance of applying maximum dimensional T′s. Where an N-by-N kernel may be used to construct the "feature space," the performance of X-space is typically eliminated.

K = (u, v) = (Φ(u), Φ(v)) = Φ(u)TΦ(v)

It displays the alternate feature area's internal product space. The dual form that has surfaced in the evolving kernel allows one to formulate a PCA mathematical equation in which the eigenvalues and eigenvectors are not solved in the \\(u)-space covariance matrix. The dot product of a single point in the converted data expressed in terms of transformed points (N points) is displayed by the N elements in column K. In the examples provided, only a few well-known kernels were suggested. Owing to feature space's direct workings, KPCA's development is constrained. After processing PCA, these data are depicted on components. To ascertain how feature space T(u) is represented to a kth principal component, Vkha has been defined as follows:

$$V^{k^T}\Phi(u) = \left(\sum_{i=1}^{N} a_{i^k}\,\Phi(u_i)\right)^T \Phi(u)$$

T(ui)TT(u) is clearly a dot product, denoted as components of a kernel K. The left portion is optimized and calculated using ak, and it is processed using the eigenvector function solver.

$N\lambda a = Ka$

where N is the number of data points, and the terms "eigen values" and "eigenvectors" refer to K's λ and a, respectively. It is defined as, eigenvectors $ak's$ are followed by normalization.

1 = (Vk)Vk

Since the u unit is made up of zero-mean in the real space, the centered feature space is not guaranteed. Since centered data is required to calculate a profitable PCA, K's initiation as K' is centralized.

$K' = K - 1_N K - K1_N + 1_N K 1_N$

where an element uses $1/N$ value in an N-by-N matrix denoted by $1N$. The kernel PCA model, as described below, is processed using the K'. A single kernel PCA caveat is illustrated. The eigen values are used in linear PCA to rank eigenvectors according to the data difference that each PCA captures. It can be applied to the KPCA process of data dimensionality removal. Therefore, it is presumed that the data variations are identical. It results from the inadequate choice of kernel scale.

DNN based classification

After the feature reduction procedure is complete, the feature values or vectors are classified. Typically, categorization is shown as a line dividing the classes, with labels applied based on the identified characteristics. DNN is used as a classification model in this instance. The DNN is characterized as an unsupervised pre-training model that employs greedy layer-wise training and is typically operated on FFNN. This method does not use a looping function as it moves data from the input layer to the output layer. One advantage of DNN classification is its minimal approach to classifying missing value feasibilities. At the unsupervised pre-training stage, it displays a single layer. Every sample of input data x = [x1, ⋯ ⋯ xN] represents a forward pass. The function where a sequence of processing layers is implied in Eq. is therefore denoted by f.

$$Z_{ij} = X_i W_{ij}; Z_j = \sum_i Z_{ij} + b_j; X_j = g(Z_j),$$

Where xi represents the input layer, xj represents the output layer, wij are the modeling parameters, and $g(Z$i) evaluates the pooling function. When layer-wise relevance propagation is applied to the classification process as described in Eq., it degrades the classification result f(x) with respect to relevance ri with input component xi.

$$f(x) = \sum_i r_i$$

Where rj < 0 indicates negative classification evidence and rj > 0 indicates positive classification evidence; otherwise, it is referred to be neutral evidence even though related attribute ri is assessed using Eq.

$$r_i = \sum_j \frac{zij}{\Sigma_i zij}.$$

A typical DNN structure is displayed in Fig.-2. The undesirable feature correlation of the input can be analyzed by the DNN. It provides a feature learning technology that is hierarchical. Consequently, DNN's primary goal is to solve the laborious functions and produce high-level abstraction.

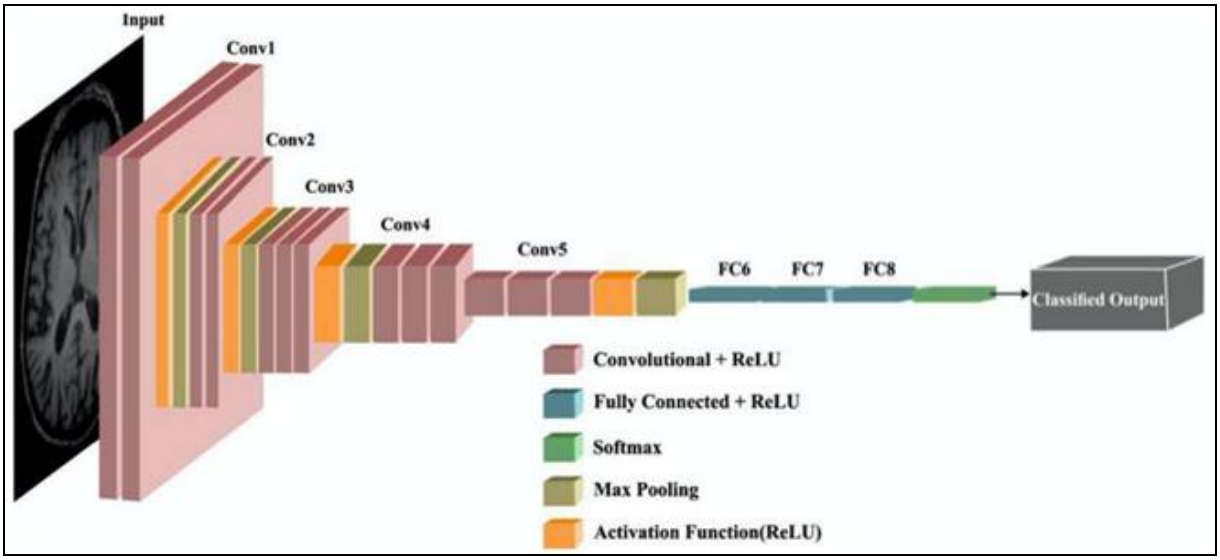

Fig 2: Architecture of deep neural networks (DNN)

Performance validation

Dataset Description: The anticipated approach's functionality is confirmed through the use of a standard dataset obtained from Kaggle. A total of 98 normal and 155 aberrant MRI brain pictures are included in the dataset. The photos range in size from 192x192 to 630x630. Fig.-3 shows a few sample test photographs.

Evaluation Measures

The newly created OS-DNN method's sensitivity, specificity, and accuracy are measured using a set of criteria.

Sensitivity: It estimates the number of positive samples correctly classified.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Specificity: It determines the ratio of negative samples that are classified correctly.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Accuracy: It calculates the amount of accurately categorized instances over the entire samples which has to be classified.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

where True positives (TP): As predicted, the instance belongs to the class, True negatives (TN): As expected, the instance does not belong in the class. False positives (FP) are instances that are anticipated to be positive even though they do not belong in the class, and false negatives (FN) are instances that are predicted to be negative even though they do belong in the class.

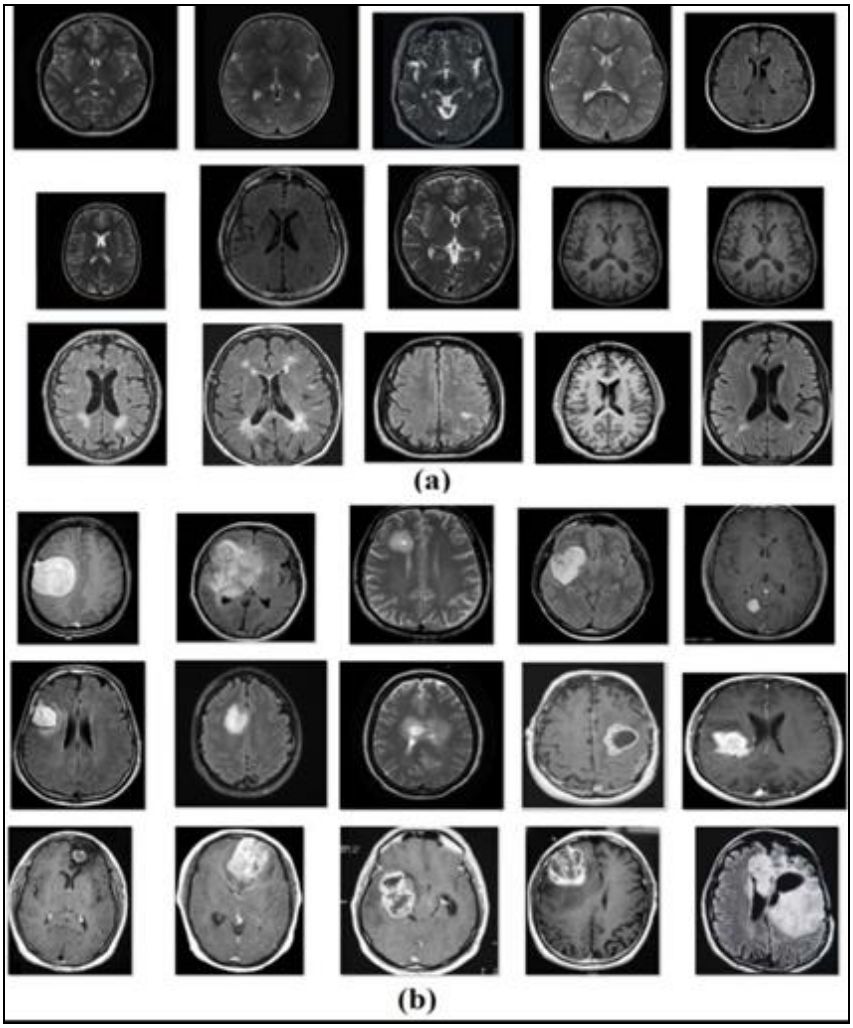

Fig 3: Sample images a) normal b) abnormal or tumor images

Results and Discussion

Fig.-4 shows an example of how the results provided by the OS-DNN model that was described can be visualized on the applied images. Figure-4 displays the original test images; Figure-4 shows the contrast-enhanced image; Figure-4 shows the skull-stripped image; this Figure displays the segmented image; and Figure displays the categorized image. The OS-DNN model successfully classified the photos into the relevant class labels, as demonstrated by the figures.

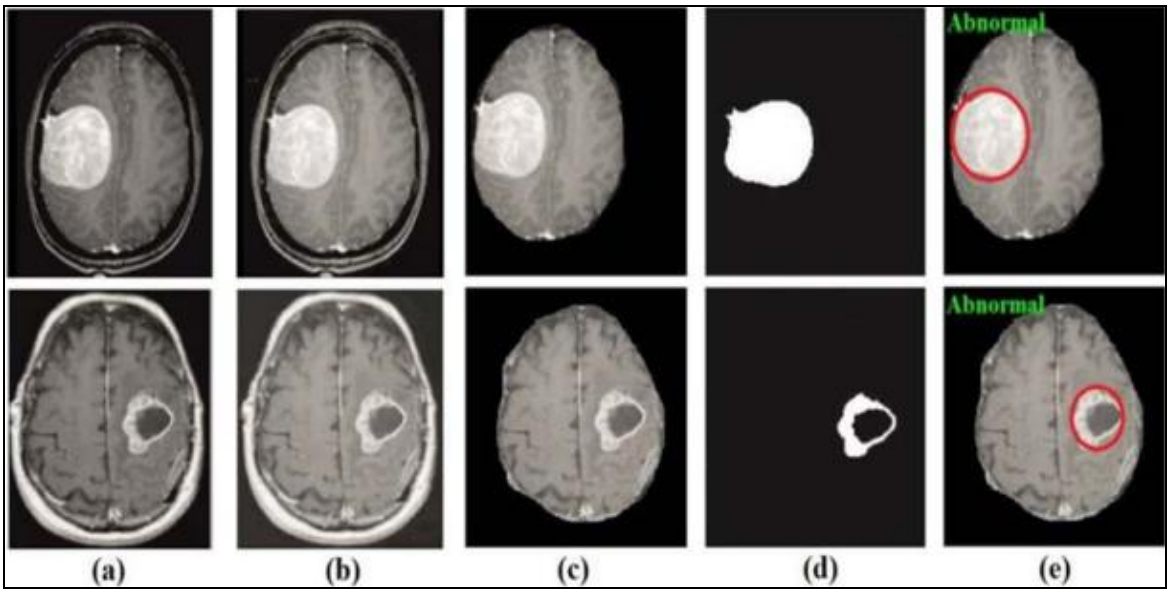

Fig 4: A) Original B) Contrast enhanced C) Skull stripped D) Segmented e) Classified

The confusion matrix of the OS-DNN model created during execution is shown in Fig-5. The picture made it quite evident that a group of 95 images out of 98 were correctly identified as normal by the suggested OS-DNN model, and 153 out of 155 images were classified as aberrant. These numbers indicate that the MRI brain images were successfully categorized by the OS-DNN model.

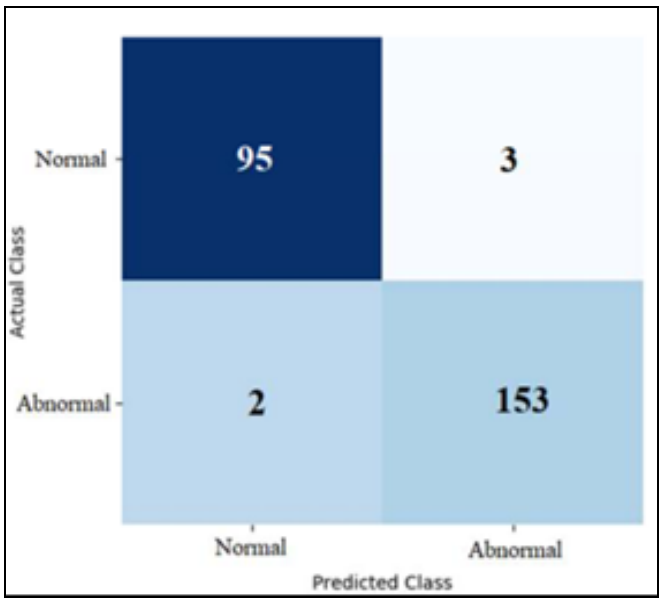

Fig 5: Confusion matrix of OS-DNN

Table-1 presents a thorough analysis of the outcomes provided by the OS-DNN and other models. The OS-DNN classification result analysis with comparison models in terms of sensitivity is displayed in Fig.-6. With a sensitivity of 80%, the k-NN model's outcomes as a classifier were ineffective, as seen in the figure-6. Subsequently, the CART model showed a marginally improved sensitivity of 88%. Subsequently, the techniques created by Anitha *et al*. (2017) [9] and Urban *et al*. (2014) [10] have demonstrated sensitivity levels of 91.2% and 92.6%, which are even higher and closer. Additionally, the sensitivity values of 94.2%, 94.3%, and 94.73% for the models created by Pereira *et al*. (2016) [11], Islam *et al*. (2013) [12], and SVM (kernel polynomial) models were shown to be mild as well as almost identical. Moreover, with a sensitivity of 95.62%, the SVM (kernel RBF) has attempted to display findings that are manageable. Furthermore, the sensitivity of 96% has been reached by both the RF and linear SVM models. Additionally, Selvapandian *et al*. (2018) [13] 's technique showed an even higher sensitivity of 96.2%. In addition, the DT model's sensitivity of 97.88% has demonstrated results that are almost ideal. Ultimately, the OS-DNN model achieved a sensitivity value of 97.94% and demonstrated better performance when compared to previous approaches.

Table 1: Result analysis of existing with proposed method in terms of sensitivity, specificity, accuracy

Methods	Sensitivity	Specificity	Accuracy
Proposed OS-DNN	97.94	98.08	98.02
SVM (kernel polynomial)	94.73	97.59	96.18
SVM (kernel RBF)	95.62	83.71	89.88
Decision tree	97.88	91.71	94.95
CART	88.00	80.00	84.00
RF	96.00	80.00	88.00
k-NN	80.00	80.00	80.00
Linear SVM	96.00	80.00	88.00
Selvapandian *et al*. (2018) [13]	96.20	95.10	96.40
Anitha *et al*. (2017) [9]	91.20	93.40	93.30
Pereira *et al*. (2016) [11]	94.20	94.40	94.60
Urban *et al*. (2014) [10]	92.60	93.00	93.30
Islam *et al*. (2013) [12]	94.30	95.10	95.90

The OS-DNN classification result analysis with related approaches in terms of specificity is implied in Fig.-6. The chart clearly shows that, by achieving the same specificity of 80%, the CART, RF, k-NN, and Linear SVM approaches have produced reasonable classifier outcomes. Subsequently, the kernel RBF (SVM) model has demonstrated a marginally superior specificity of 83.71%. Furthermore, the DT made an effort to demonstrate noteworthy outcomes with a 91.71% specificity.

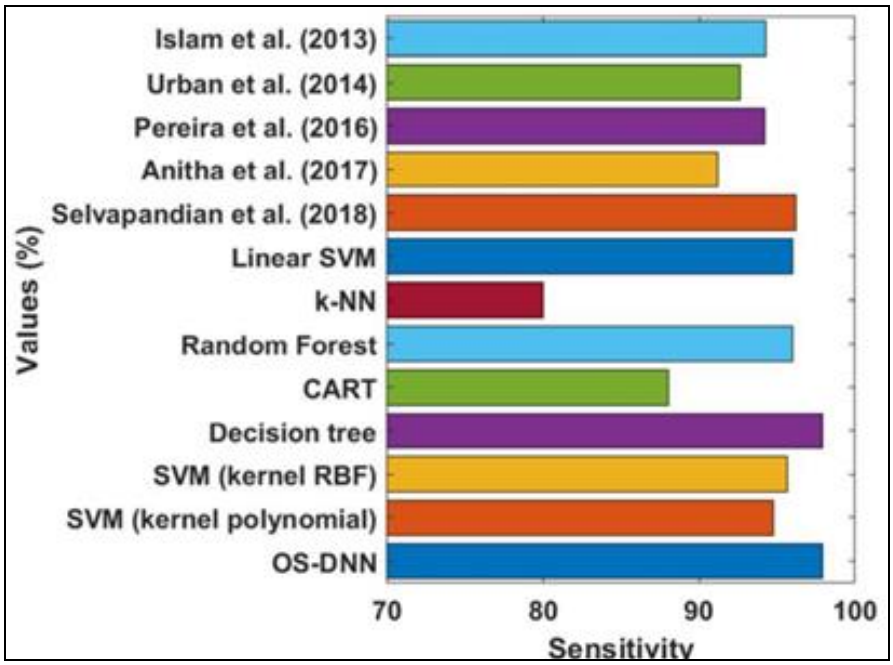

Fig 6: Sensitivity analysis of OS-DNN with other models

After then, the models used by Anitha *et al*. (2017) [9] and Urban *et al*. (2014) [10] shown improved and close specificity values of 93.40% and 93%. Conversely, the specificity value of the model reported by Pereira *et al*. (2016) [11] is 94.40%. Furthermore, the models proposed by Islam *et al*. (2013) [12] and Selvapandian *et al*. (2018) [13] have attempted to suggest realistic outcomes and a comparable specificity of 95.10 percent. With a specificity of 97.59%, the SVM (kernel polynomial) model has shown closer ideal results in accordance with this. Finally, the OS-DNN approach achieved a higher specificity value of 98.08%, which is superior than previous models.

The accuracy of the OS-DNN classification result analysis with related models is shown in Fig.-7. The k-NN model was shown to produce subpar classifier results, with an accuracy of 80%, according to the figure-7. Furthermore, the CART method has demonstrated a noteworthy 84% accuracy rate. Conversely, both the RF and Linear SVM models have demonstrated superior and comparable accuracy levels of 88%. Furthermore, the SVM has demonstrated a noteworthy accuracy of 89.88%. Concurrently, the methodologies employed by Anitha *et al*. (2017) [9] and Urban *et al*. (2014) [10] have demonstrated even more similar accuracy rates of 93.30%.

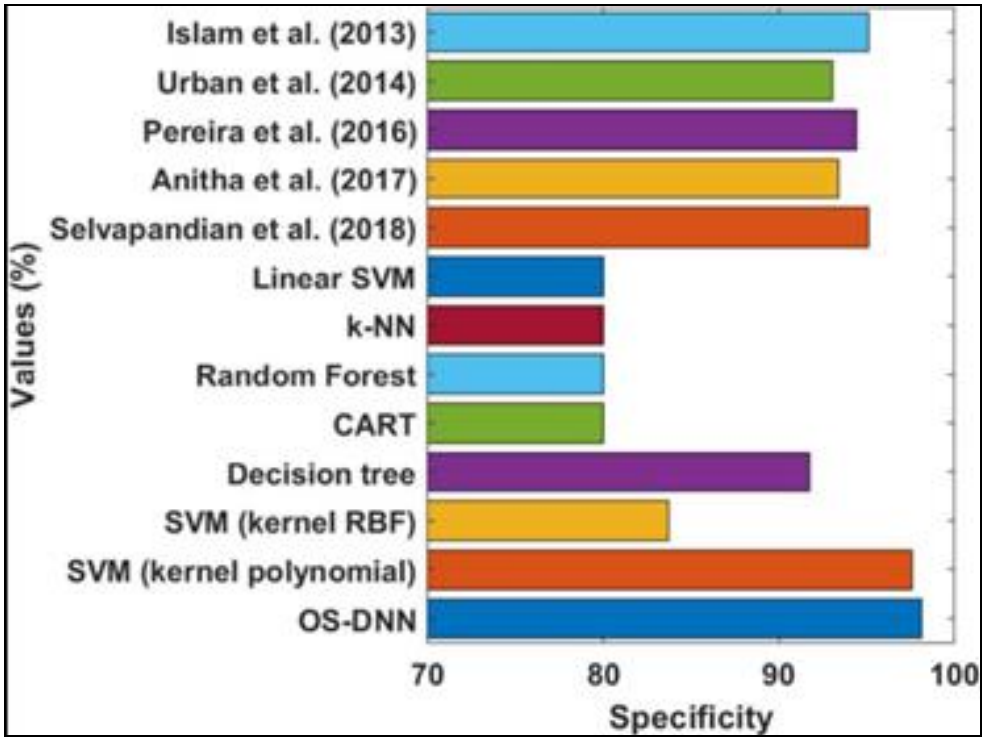

Fig 7: Specificity analysis of OS-DNN with other models

Furthermore, with an accuracy of 94.60% and 94.95%, respectively, the models used by Pereira *et al*. (2016) [11] and DT models tried to indicate appropriate outcomes. Furthermore, the technique used by Islam *et al*. (2013) [12] achieved a 95.90% accuracy rate. Furthermore, the kernel polynomial (SVM) method has demonstrated superior accuracy of 96.18%. Accordingly, the Selvapandian *et al*. (2018) [13] architecture has demonstrated near-optimal performance, with an accuracy rate of 96.40%. Finally, compared to similar models, the OS-DNN model has demonstrated ideal outcomes with an accuracy value of 98.02%.

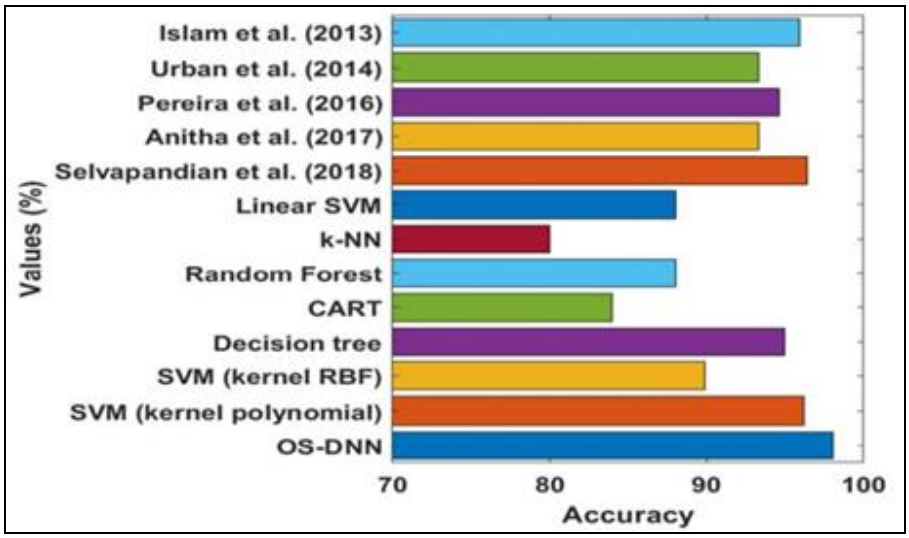

Fig 8: Accuracy analysis of OS-DNN with other models

As can be seen from the aforementioned tables and figures-9, the OS-DNN model is a useful tool for BT diagnosis and categorization from MRI brain images.

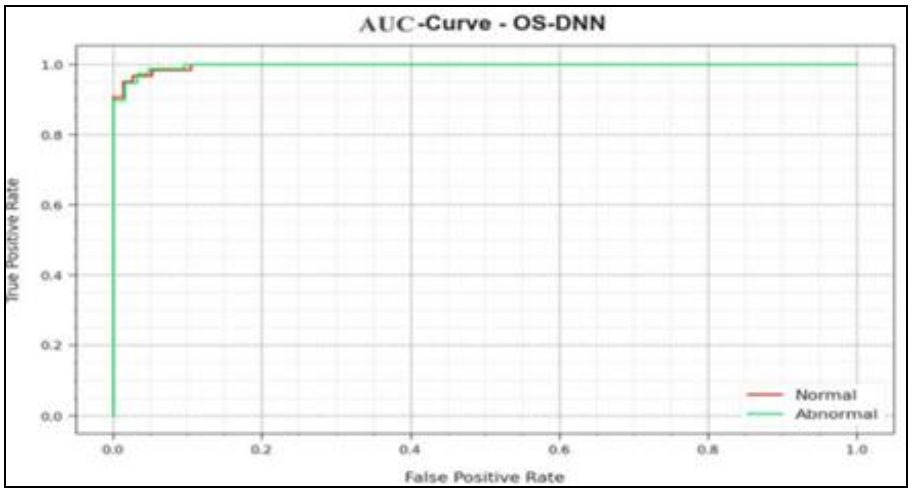

Fig 9: AUC-curve for OS-DNN model

AUC Curve for the comparison performance study of the suggested work is shown in fig.-9.

The suggested OS-DNN model includes the following features: DWT based feature extraction, KPCA based feature reduction, Kapur's thresholding based segmentation, BF based noise removal, CLAHE based contrast enhancement, DNN based classification for BT classification procedure, and GWO based threshold value selection.

Conclusion

To the best of our knowledge, there has never been a published version of the OS-DNN model for BT classification. The Kapur's thresholding approach's threshold value selection is based on a thorough search through all potential threshold values, which can be computationally costly. The OS-DNN model uses the architecture of the GWO algorithm, which can effectively search over a wide solution space to find the ideal threshold value, to address this problem. When compared to exhaustive search methods, these algorithms can drastically cut down on the computational time needed for threshold selection. The scientific community and healthcare facilities have focused their attention on the most effective ways to detect and classify BT. Novel BT detection and categorization utilizing the OS-DNN model have been described in this paper. Firstly, the suggested OS-DNN model is preprocessed using noise reduction, contrast

improvement, and skull stripping in order to boost the classification performance. Subsequently, segmentation based on GWO-KT, feature extraction based on DWT, feature reduction based on KPCA. The suggested MOAOA-FDL technique may eventually be expanded to include the formulation of an instance segmentation strategy based on DL. Additionally, large-scale real-time datasets were used to test the provided approach in the future.

References

1. Afshar P, Mohammadi A, Plataniotis KN. Bayes Cap: A bayesian approach to brain tumor classification using capsule networks. IEEE Signal Process Lett. 2020;27(12):2024-2028.
2. Hirahara D. Preliminary assessment for the development of CAD system for brain tumor in MRI images utilizing transfer learning in Xception model. In: IEEE 8th Global Conference on Consumer Electronics (GCCE); c2019. p. 922-924.
3. Deepa R, Sam Emmanuel WR. MRI brain tumor classification using cuckoo search support vector machines and particle swarm optimization based feature selection. In: 2nd International Conference on Trends in Electronics and Informatics (ICOEI); c2018. p. 1213-1216.
4. Dongyuan Wu, Yi Din, Mingfeng Zhang, Qiqi Yang, Zhiguang Qin. Multi-features refinement and aggregation for medical brain segmentation. IEEE Access. 2020;8(6):57483-57496.
5. Han Y, Zhang Z. Notice of violation of IEEE publication principles: Deep learning assisted image interactive framework for brain image segmentation. IEEE Access. 2020;8(11):117028-117035.
6. Jain M, Singh V, Rani A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm Evol Comput. 2019;44(2):148-175.
7. Soumik MFI, Hossain MA. Brain tumor classification with inception network based deep learning model using transfer learning. In: IEEE Region 10 Symposium (TENSYMP); c2020. p. 1-14.
8. Zhou C, Ding C, Wang X, Lu Z, Tao D. One-pass multi-task networks with cross-task guided attention for brain tumor segmentation. IEEE Trans Image Process. 2020;29(5):4516-4529.
9. Anitha K, Neelamegam K, Kanagaraju P. Confirmatory Analysis on Morale of Women Employees in Unorganised Sector on Work Outcomes. International Journal of Advanced Scientific Research & Development (IJASRD). 2017;4(7):76-85.
10. Childers DL, Pickett ST, Grove JM, Ogden L, Whitmer A. Advancing urban sustainability theory and action: Challenges and opportunities. Landscape and urban planning. 2014;125:320-328.
11. Teixeira J, Pereira OM. The determinants of capital structure of Portuguese firms. CEEAplA-A-Working Paper Series; c2016. p. 1-32.
12. Islam MA. Modeling univariate Volatility of Stock Returns using stochastic GARCH models. Evidence from 4 Asian Markets. Australian Journal of Basic and Applied Sciences. 2013;7(11):294-303.
13. Selvapandian A, Manivannan K. Fusion based glioma brain tumor detection and segmentation using ANFIS classification. Computer methods and programs in biomedicine. 2018;166:33-38.