ANN Progress Report
Deep Learning Based Brain Tumor Detection

Bekir Emirhan Akay (19290286), Mehmet Alpay(20290310)
Fatih Yalçın (20290372), Murat Tatlı (20290366)

December 7, 2023

1 Problem Description

Brain tumor detection is the process of identifying the presence of a tumor in the brain. It is typically done using medical imaging techniques such as magnetic resonance imaging (MRI). Detecting brain tumors can be difficult due to the variations in tumor location, shape, and size. Machine learning techniques have emerged as a promising approach for automated brain tumor detection, leveraging the power of artificial intelligence to analyze medical images accurately and efficiently. We are interested in this area that is very useful for humanity because every year. In 2020, an estimated 251,329 people worldwide died from primary cancerous brain and CNS tumors.[1] Thus, we want to contribute to early diagnosis of the tumors.

2 Related work

This section presents an examination of various algorithms employed for the detection of brain tumors. Several methodologies have been explored, including the successful application of the supervised method, specifically random forests, for brain tumor segmentation [2]. Tumor segmentation has also been achieved through the implementation of the Gaussian mixture model (GMM) [3]. Notably, the utilization of morphological/contextual features has demonstrated enhanced results in the detection of brain tumors [4], while the Markov Random Field (MRF) has been employed for precise lesion region segmentation [5].

The efficacy of deep learning approaches lies in their ability to automatically extract highly discriminative features hierarchically. A five-layer Convolutional Neural Network (CNN) model is employed for classifying enhanced, non-enhanced, and core tumor regions [6]. Further advancements include the extraction of different patches from original images, which are then subjected to CNN for the segmentation of High-Grade Glioma (HGG) and Low-Grade Glioma (LGG) cases [7].

Diverse methods such as 2D CNN, U-Net [8], CNN [9], Deep Neural Networks (DNN) [10], multi-scale 3D CNN [11], multi-fractal Detrended Fluctuation Analysis (MFDFA) [12], pairwise affinity and super pixel [13], hierarchical classification [14], local independent projection-based classification [15], extremely randomized trees [16], generative models [17], random field (RF) [18], fully convolutional neural networks (FCNNs) [19], and cascaded CNN [20] have been employed for brain tumor detection. Although substantial work has been conducted in the literature, there exists a gap in achieving optimal results across all performance metrics. Consequently, the proposed model introduced in this study has demonstrated commendable results in terms of various performance metrics on five challenging datasets. This success can be attributed to the innovative fused CNN model presented for accurate brain tumor diagnosis. For a comprehensive overview of existing deep learning methods, refer to Figure 1 [21].

Ref	Year	DL methods	BRATS Datasets	Performance measures (DSC)		
				Complete	Core/non-enhance	Enhance
Zikic, Ioannou, Brown, and Criminisi (2014)	2014	CNN	2013	83.7 ± 9.4	73.6 ± 25.6	69.0 ± 24.9
Dvořák and Menze (2015)	2015	CNN	2014	81 ± 15	79 ± 13	81 ± 11
Pereira et al. (2016)	2016	CNN	2013	0.88	0.83	0.77
Havaei (2017)	2017	DNN	2013	0.81	0.72	0.58
Dong et al. (2017)	2017	U-Net CNN	2015	0.86	0.86	0.65
Kamnitsas (2017)	2017	3D-CNN + RF	2015	0.84	0.66	0.63
Zhao et al. (2018)	2018	FCNN model	2013	0.70	0.62	0.64
Abd-Ellah, Awad, Khalaf, and Hamed (2018)	2018	CNN	2013	0.87	–	–
Hussain, Anwar, and Majid (2018)	2018	DNN	2013	0.87	0.89	0.92
			2015	0.86	0.87	0.90
Chen, Ding, and Liu (2019)	2019	Dual-force CNN	2015	0.85	0.70	0.63
			2017	0.89	0.73	0.73

Figure 1: Related deep learning techniques [21]

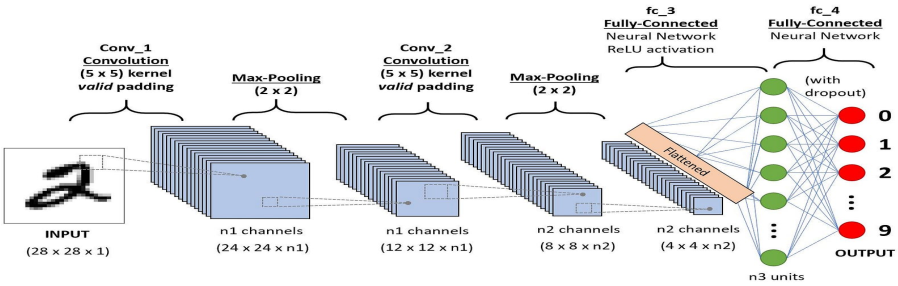


Figure 2: Basic structure of CNN[24]

3 Methodology

3.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) stands as a specialized variant of neural networks with widespread applications in various domains, encompassing image and video recognition, recommender systems, image classification, image segmentation, medical imaging, linguistic analysis, brain-computer interfaces, and financial time series analysis. In medical disease detection, CNN architectures are one of the best solutions in the literature. CNN gives good results because the majority of medical disease detection, datasets consist of image files such as jpg, png,dcm, etc. and CNN networks have good coverage for image formats. A basic example of the CNN network is given in Figure X. In this progress report, we used 8 different CNN architectures to train the model. Moreover, we used the F1 score, recall, precision, and accuracy metrics to measure the performance. Especially F1 score is a critical metric to us because it calculates all possible worst scenarios (false positives, false negatives, etc). We described some of the CNN architecture that is used in our experiments. Then, we explain which improvements that we made. The basic structure of the CNN models are shown in Figure-2.

3.1.1 MobilenetV2

MobileNetV2 is a type of neural network that is optimized for mobile devices. It is designed to be lightweight and efficient while providing high accuracy. The network is based on an inverted residual structure, which means that the residual connections are between the bottleneck layers. Originally, Residual layers were used in ResNet architecture MobileNetV2 uses an inverted connected version of this, and with a little bit of loss of accuracy, it can cover the data very fast. Also, bottleneck layers get important features from the activation layer's result to summarize data. This helps to reduce the computational cost of the model. The intermediate expansion layer uses depthwise convolutions to filter features, which provides a source of non-linearity. The architecture of MobileNetV2 consists of an initial fully convolutional layer with 32 filters, followed by 19 residual bottleneck layers.

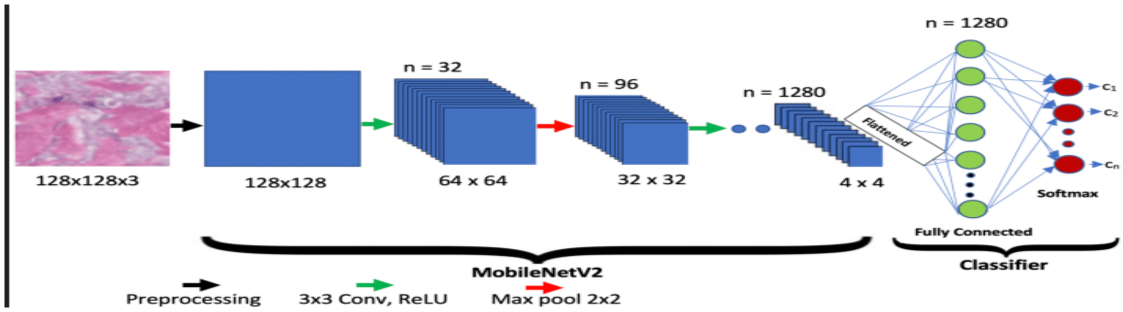


Figure 3: MobilenetV2 Architecture[25]

3.1.2 DenseNet121

DenseNet-121 is a type of neural network that is designed to improve the training of deep neural networks. It is based on the idea of densely connected convolutional networks, where each layer is connected to every other layer in a feed-forward fashion. This helps to reduce the vanishing gradient problem that arises when the number of layers in the network increases. The architecture of DenseNet-121 consists of a series of dense blocks, each containing multiple layers. The feature maps of all the previous layers are concatenated and used as inputs for each layer in the block. This allows for feature reuse and reduces the number of parameters in the model. The architecture also includes transition layers, which are used to reduce the number of channels in the feature maps.

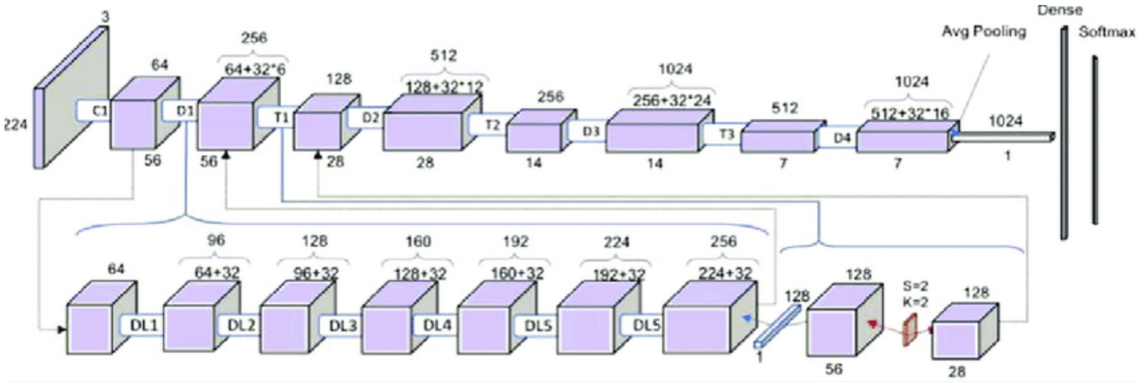


Figure 4: Densenet121 Architecture[26]

3.1.3 ResNet152V2

ResNet152V2 is a convolutional neural network architecture that is designed to improve the training of deep neural networks. It is based on the idea of residual connections, where each layer is connected to every other layer in a feed-forward fashion. It uses Residual layers to reduce the vanishing gradient problem that arises when the number of layers in the network increases. The architecture of ResNet152V2 consists of a series of residual blocks, each containing multiple layers. The feature maps of all the previous layers are added to the output of each layer in the block. This allows for feature reuse and reduces the number of parameters in the model. The architecture also includes transition layers, which are used to reduce the number of channels in the feature maps.

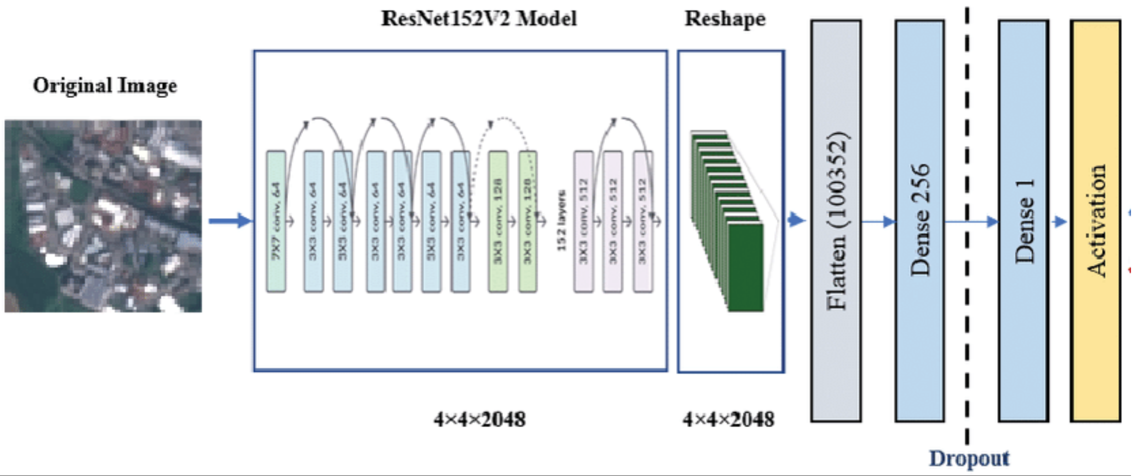


Figure 5: ResNet152V2 Architecture[27]

For these models, we modified some parts of the architecture. Firstly, we increase the probability of dropout to 30% because when we train, we detect overfitting situations. We added L2 regularization to overcome the overfit scenario. Moreover, we used some data augmentation techniques to get more data to train data. For example, we used random rotation, shearing, and flips to augment the data. Lastly, to get more parameters to generalize the model we increase the number of fully connected layers. By using these adjustments performance metrics increase dramatically. An example of data augmentation is shown below.

4 Experimental Evaluation

In our proposed methods we used the same hyperparameters for controlled experiments. In hyperparameters, we define some rules for each model. For example, for the optimizer, we prefer to use SGD with a learning rate of 10eˆ3. We set the number of epochs as 5. This may be a very low number for image classification tasks but we use transfer learning to reduce the required epoch number. The weights used in our models are taken from imagenet pre-trained weights. For each CNN architecture, we used data augmentation techniques such as shearing for cutting random parts of the image to learn more efficiently, random rotation, and flips. This augmentations has a very vital role for the task's performance because we generate more different data to feed our model. In the original dataset approximately there are 22k images but with augmentation, datasets approximately doubled. Increased data make stronger our model. Moreover, to reduce overfitting risk we added L2 regularization. Our training results for different CNN architectures are shown below.

Model Name	Train Loss	Train Acc	Train Precision	Train Recall	Train F1 Score
MobileNetV2	0.0805	0.9717	0.9763	0.9687	0.9595
DenseNet121	0.0572	0.9807	0.9838	0.9783	0.9736
Xception	0.1716	0.9478	0.9580	0.9383	0.9285
ResNet50V2	0.0916	0.9682	0.9730	0.9636	0.9557
ResNet101V2	0.0531	0.9799	0.9832	0.9775	0.9736
ResNet152V2	0.0465	0.9827	0.9858	0.9805	0.9757

In the shown table ResNet152V2 gives the best results for training purposes. DenseNet121 and ResNet101V2 also have good performance. Validation scores are shown below.

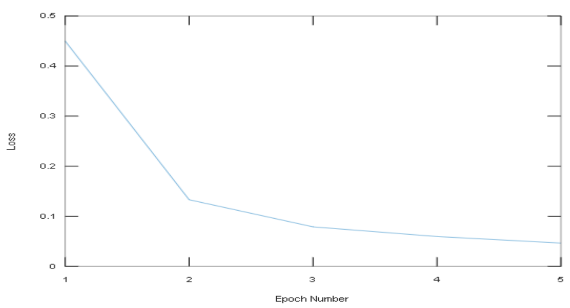
Model Name	Val Loss	Val Acc	Val Precision	Val Recall	Val F1 Score
MobileNetV2	0.3617	0.8843	0.8915	0.8810	0.8807
DenseNet121	0.0859	0.9734	0.9734	0.9728	0.9636
Xception	0.2807	0.9184	0.9312	0.8967	0.8936
ResNet50V2	0.1973	0.9374	0.9424	0.9334	0.9220
ResNet101V2	0.1390	0.9590	0.9599	0.9587	0.9440
ResNet152V2	0.0905	0.9711	0.9724	0.9708	0.9604

In the shown table Densenet121 gives the best results for the validation set. However, ResNet152V2 has very near results. We think that Densenet121, ResNet152V2, and ResNet101V2 are the best models. We write the test results below.

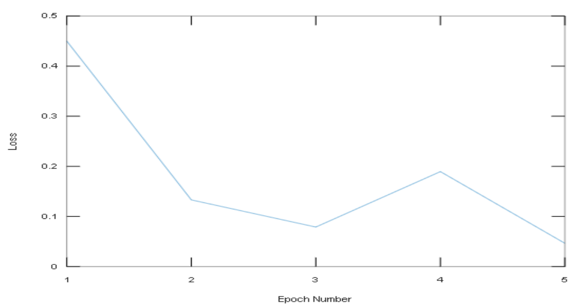
Model Name	Test Loss	Test Acc	Test Precision	Test Recall	Test F1 Score
MobileNetV2	0.4298	0.8561	0.8739	0.8433	0.8173
DenseNet121	0.3907	0.8895	0.8895	0.8836	0.8054
Xception	0.3734	0.8830	0.8918	0.8725	0.8100
ResNet50V2	0.2227	0.9285	0.9325	0.9239	0.9065
ResNet101V2	0.2316	0.9439	0.9448	0.9433	0.9250
ResNet152V2	0.1020	0.9718	0.9731	0.9708	0.9561

When we look at the test results, the best model is Resnet152V2 with our fine-tuning methods. Thus, we will focus on ResNet152V2 to improve and add new techniques.

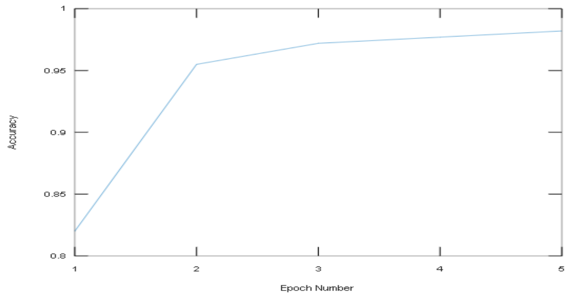
We provide Resnet152V2 loss and accuracy metrics graphics to analyze overfit/underfit conditions. Graphs are shown below.



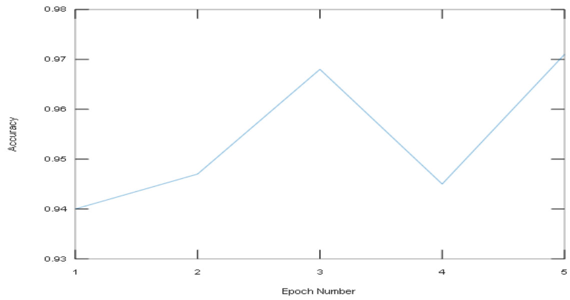
(a) Train Loss



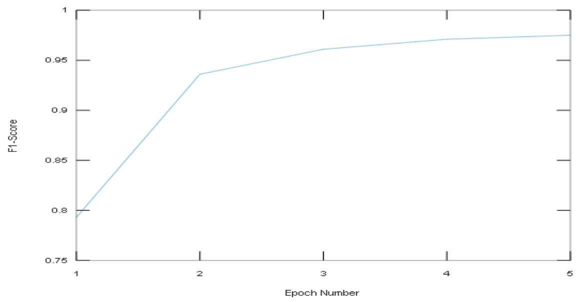
(b) Validation Loss



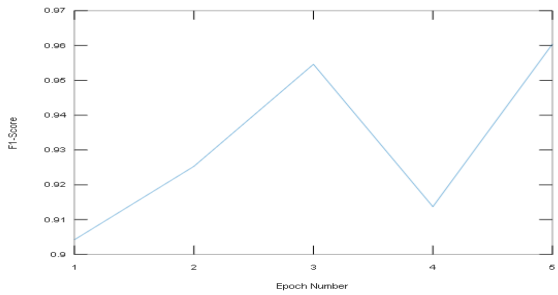
(c) Train Accuracy



(d) Validation Accuracy



(e) Train F1 Score



(f) Validation F1 Score

5 Baseline Results

We compare our proposed methods with a baseline model which is VGG-16. VGG-16 is one of the oldest CNN architecture in the literature. Therefore, we compared our results with pre-trained VGG without making any modification. The results are shown below.

Model Name	Train Loss	Train Accuracy	Train Precision	Train Recall	Train F1 Score
VGG16	0.2538	0.8954	0.9051	0.8849	0.8320

Model Name	Validation Loss	Validation Accuracy	Validation Precision	Validation Recall	Validation F1 Score
VGG16	0.2761	0.9007	0.9076	0.8921	0.8246

Model Name	Test Loss	Test Accuracy	Test Precision	Test Recall	Test F1 Score
VGG16	0.3238	0.8790	0.8852	0.8751	0.7825

References

[1] Attention Is All You Need https://proceedings.neurips.cc/paper-files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[2] Deep Learning for Smart Healthcare—A Survey on Brain Tumor Detection from Medical Imaging (sensors-22-01960.pdf)

[3] Doi, K. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. Comput. Med. Imaging Graph. Off. J. Comput. Med. Imaging Soc. 2007, 31, 198–211. [CrossRef] [PubMed]

[4] Geremia, E., Menze, B. H., Clatz, O., Konukoglu, E., Criminisi, A., & Ayache, N. (2010). Spatial decision forests for MS lesion segmentation in multi-channel MR images. In International conference on medical image computing and computer-assisted intervention (pp. 111–118). Springer.

[5] Van Leemput, K., Maes, F., Vandermeulen, D., & Suetens, P. (1999). Automated model-based tissue classification of MR images of the brain. IEEE Transactions on Medical Imaging, 18(10), 897–908.

[6] Rao, A., Ledig, C., Newcombe, V., Menon, D., & Rueckert, D. (2014). Contusion segmentation from subjects with traumatic brain injury: A random forest framework. In 2014 IEEE 11th international symposium on biomedical imaging (ISBI) (pp. 333–336). IEEE.

[7] Mitra, J. et al. (2014). Lesion segmentation from multimodal MRI using random forest following ischemic stroke. NeuroImage, 98, 324–335

[8] Kleesiek, J., Biller, A., Urban, G., Kothe, U., Bendszus, M., & Hamprecht, F. (2014). Ilastik for multi-modal brain tumor segmentation. Proceedings MICCAI BraTS (Brain Tumor Segmentation Challenge), 12–17.

[9] Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2015). Deep convolutional neural networks for the segmentation of gliomas in multi-sequence MRI. In BrainLes 2015 (pp. 131–143). Springer.

[10] Dong, H., Yang, G., Liu, F., Mo, Y., & Guo, Y. (2017). Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks. In Annual conference on medical image understanding and analysis (pp. 506–517). Springer.

[11] Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2016). Brain tumor segmentation using convolutional neural networks in MRI images. IEEE Transactions on Medical Imaging, 35(5), 1240–1251.

[12] Havaei, M. et al. (2017). Brain tumor segmentation with deep neural networks. Medical Image Analysis, 35, 18–31.

[13] Kamnitsas, K. et al. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Medical Image Analysis, 36, 61–78.

[14] Reza, S. M., Mays, R., & Iftekharuddin, K. M. (2015). Multi-fractal detrended texture feature for brain tumor classification. In Medical imaging 2015: Computer-aided diagnosis (pp. 941410). International Society for Optics and Photonics.

[15] Wu, W., Chen, A. Y., Zhao, L., & Corso, J. J. (2014). Brain tumor detection and segmentation in a CRF (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features. International Journal of Computer Assisted Radiology and Surgery, 9(2), 241–253.

[16] Bauer, S., Fejes, T., Slotboom, J., Wiest, R., Nolte, L.-P., & Reyes, M. (2012). Segmentation of brain tumor images based on integrated hierarchical classification and regularization. MICCAI BraTS work-shop. Nice: Miccai Society

[17] Huang, M., Yang, W., Wu, Y., Jiang, J., Chen, W., & Feng, Q. (2014). Brain tumor segmentation based on local independent projection-based classification. IEEE Transactions on Biomedical Engineering, 61 (10), 2633–2645

[18] Goetz, M., Weber, C., Bloecher, J., Stieltjes, B., Meinzer, H.-P., & Maier-Hein, K. (2014). Extremely randomized trees based brain tumor segmentation. In Proceeding of BRATS challenge-MICCAI (pp. 006–011).

[19] Kwon, D., Shinohara, R. T., Akbari, H., & Davatzikos, C. (2014). Combining generative models for multifocal glioma segmentation and registration. In International conference on medical image computing and computer-assisted intervention (pp. 763–770). Springer.

[20] Tustison, N. J. et al. (2015). Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with ANTsR. Neuroinformatics, 13(2), 209–225.

[21] Zhao, X., Wu, Y., Song, G., Li, Z., Zhang, Y., & Fan, Y. (2018). A deep learning model integrating FCNNs and CRFs for brain tumor segmentation. Medical Image Analysis, 43, 98–111.

[22] Havaei, M., Dutil, F., Pal, C., Larochelle, H., & Jodoin, P.-M. (2015). A convolutional neural network approach to brain tumor segmentation. In BrainLes 2015 (pp. 195–208). Springer

[23] Saba, T., Mohamed, A. S., El-Affendi, M., Amin, J., & Sharif, M. (2020). Brain tumor detection using fusion of hand crafted and deep learning features. Cognitive Systems Research, 59, 221-230.

[24] https://www.cancer.net/cancer-types/brain-tumor/statistics

[25] https://www.freecodecamp.org/news/want-to-know-how-deep-learning-works-heres-a-quick-guide-for-everyone-1aedeca88076/

[26] https://www.researchgate.net/Figure/The-proposed-MobileNetV2-network-architecture

[27] https://www.researchgate.net/figure/The-architecture-of-DenseNet121

[28] https://www.researchgate.net/ResNet152V2-Architecture