Brain Tumor Detection through MRI Images

Tiago Alvim - 95584
Universidade de Aveiro
DETI
Aveiro, Portugal
tiago.alvim@ua.pt

Vasco Costa - 97746
Universidade de Aveiro
DETI
Aveiro, Portugal
vascovc@ua.pt

Applications of Artificial Intelligence
2022/2023 - 41019 - AIA
Pétia Georgieva

***Abstract*—Tumors are abnormal masses of tissues that form when cells grow and divide more than they should being. Magnetic resonance imaging, MRI, is one of the most common techniques of imagiology.**

Convolutional Neural Networks, CNNs, are powerful tools when in respect to deep neural networks, with its primarily use being in image-driven problems. With that said CNN models were used on a MRI image dataset for comparison and evaluation on their possible use in the medical field. Namely a custom approach and by transfer learning with VGG19 and Inception V3 models pretrained on the *ImageNet* dataset. Hyperparameter search for the parameter of `loss` and the `optimizer` was implemented. Revealing how important data augmentation can prove in some cases. Additionally, the comparison to related works showed that there still exist misconceptions when in relation to the metrics being used, embellishing the result obtained.

The results obtained through transfer learning show that there is a place for the use of this methods in the medical field for assisting in the detection to better help the people.

***Index Terms*—Health Care, Brain Tumor, Machine Learning, Classification, CNN-Keras**

I. Introduction

[1] Tumors are abnormal masses of tissues that form when cells grow and divide more than they should being either benign, not of immediate concern, or malignant, dangerous [1]. Requiring medical attention with health providers that have knowledge on the diverse types known with a key importance on timing, since detection to intervention [2]. Presenting a considerable impact on the life of the patients, have it be in the physical or psychosocial aspects with different ways of measuring such impairment [3].

One of the most common techniques applied currently consists on the use of high-resolution structural imaging through physiologic imaging. As it is a non-evasive method its use to plan and monitor the patients condition is easier to accompany in the process of treatment [4]. Despite existing multiple diverse methods for imagiology an industry standard is Magnetic Resonance Imaging, MRI, it consists on the use of the body's natural magnetic properties to produce detailed images for any part of the body using hydrogen nucleus, protons. In the presence of a magnetic field with the incidence of radio wave frequencies the different tissues, with different contents of water and fat, resonate uniquely making it possible to visualise any differences [5]. Some cases also require the use of contrast an intravenous substance, *GBCAs*. As they increase the quality of the pictures by making it more clear have it be the tumor itself, veins or some organs blood supply. Leading to a easier process in recognizing recent developing tumors that could be missed by traditional methods. [6]

In the recent years the amount of data has increased drastically through the development of sensors and the internet. Leading to the process of machine learning, where the computer has the ability to improve itself. Where the topic of computer vision has also shown more demand to connect both through pattern recognition on image identification [7]. Extracting crucial information in multiple domains, from surveillance systems to healthcare [8].

In medicine there exist already extensive datasets but without the proper contribute to clinical care in oppose to other areas where it is common practice [9]. However, through the use of Deep Neural Networks, DNNs, fully automatic efficient systems can learn features unique to brain tumor segmentation [10].

In this work it is intended to study brain tumor presence through machine learning on MRI images.

II. Convolutional Neural Network

Convolutional Neural Networks, CNN, have become one of the most popular and powerful tools when in respect to deep neural networks. This is due to them being able to deal well with large data with the ability to recognize patterns more effectively than other types of machine learning algorithms. With its primarily use being in image-driven problems. This method gets its name from the mathematical operations between matrices. It is common to be in the presence of multiple layers with different functions each [11], [12].

Typically input data gets processed by a series of blocks with transformations, called forward propagation. The blocks can include *Convolution Layers* where data is extracted by a combination of linear and nonlinear operations, a convolution operation and an activation function. Convolution is an operation where a kernel is applied across the input, an array of numbers, named a tensor. Done in steps like it is presented in Fig. 1. After this process the data can be passed to an *Activation Function*. Also present can be a *Pooling Layer* to down-sample the dimensionality of the feature maps to decrease the number of trainable parameters. *Max pooling* can also be considered to accept the maximum value in

[1]The work load per student is detailed in `work_division.pdf` provided with the project

each patch and discard the other values. Through *Average Pooling* instead of considering the maximum value of the patch, the average is taken. The output of the feature maps is then flattened, transformed into a one-dimensional array and connected to fully connected layers, dense layers. These dense layers connect each input to an output by a trainable weight with appropriate activation functions according to the task at hand. The training process of a network is the process of reducing the loss through the change of the networks parameters such as kernels and weights, so that the predictions correspond to the labeled data [13].

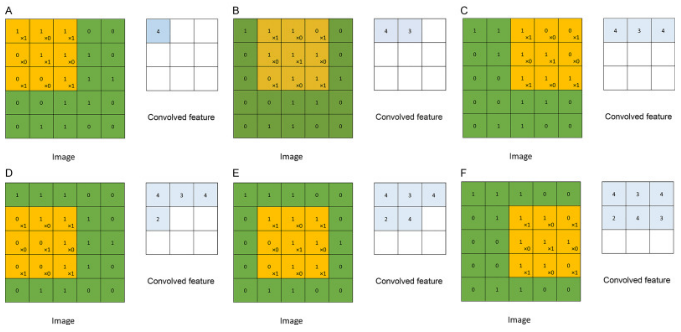


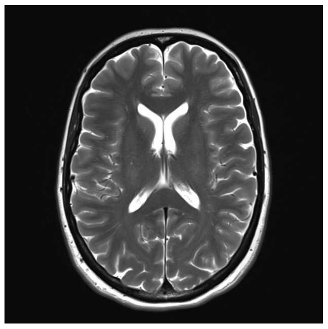
Fig. 1: Representation of the convolution operation. Figure by [14]

III. The Dataset

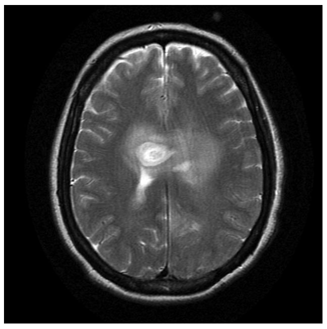
To perform this task the dataset available in [15] was chosen. It consists of three folders, but only two are relevant as the files are repeated. The 253 files are then divided in two sets, two folders, according to the condition, `yes` or `no` if in the presence of a tumor or not, respectively.

According to the dataset author, Navoneel Chakrabarty, the pictures were retrieved from Google Images and with no regard to the type of tumor just if on the presence of one, dating back to earlier than 2018.

In Fig. 2 it is possible to visualize two examples of the pictures presented in the dataset with Fig. 2a referring to a non tumours brain opposed to Fig. 2b.



(a) No



(b) Yes

Fig. 2: Two files presented in the dataset, on the left a non tumorous brain and on the right a tumorous one.

A. Data preprocessing

Firstly the images were examined and it was clear that the dataset is unbalanced as it is possible to verify by visualizing in Fig. 3. With class `no` having 98 elements and class `yes` 155.

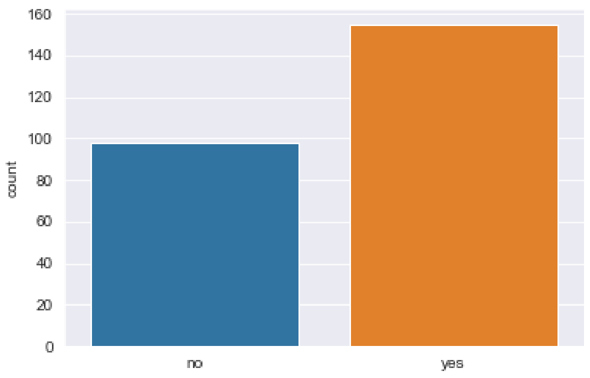


Fig. 3: Representation of the number of pictures labeled as `no` and `yes` in the dataset.

With each picture having an unique resolution, ranging from 150 by 198 to 1920 by 1080 in the `no` category and the `yes` having files ranging from 178 by 249 to 1275 by 1427. Making it a must to reshape into a standardized image size done with the use of `opencv` [16]. As these are MRI images it is also possible to read them in grey scale as it is how they were captured originally, reducing the number of channels to analyse making it faster to compute.

After they were read and normalized, divided by 255. The pictures are divided into a train set and a test set of 70%, 30% each, respectively.

In order to have a wider range of samples to train and prevent overfitting the method of augmenting pictures was used on the train set portion, with the tool `ImageDataGenerator` from `keras.preprocessing.image` [17]. In some pictures the procedure of rotating them slightly and flipping both horizontally and vertically. Since a head can be approximately considered as a circumference, the method should be well suited to recognize it. Zooming in on some pictures just slightly to simulate a not so good imaging procedure and then shifting it from side to side both horizontally and vertically. The values used for the ranges could also be considered as values for hyperparameters.

IV. Custom Method

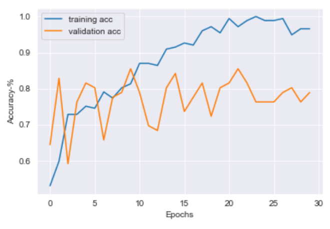
[2] As a first approach to this task a custom solution was developed to better understand the process behind deep neural network development to further study with examples from the literature that are more advanced/complex requiring higher computational power.

[2]Code available in `custom_model.ipynb`

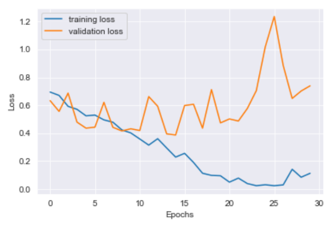
The development of this model took base on the exercise done in class with some tweaking by adding some layers through trial and error using the preprocessing described earlier on images read in grey-scale on a resolution of 140 by 140 pixels.

By using 4 `conv2D` layers, with the first having 64 filters and the last 16 with the middle ones having 32. These layers were intercalated by the use of `MaxPool2D` layers. Finishing off with a `Dropout` layer and a `Flatten` one before proceeding to 4 fully connected layers, `Dense`, with the first three having the sizes of 256, 128 and 64 in order with the *relu* activation function finishing in the last one with one output with the *sigmoid* activation function to provide a result. Leading to a model with 336,593 total parameters all being trainable.

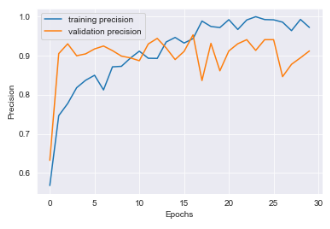
Resulting on the graphs presented on Fig. 4. The values here portrayed suffer from very abrupt changes showing too big of a step on the training data fitting procedure.



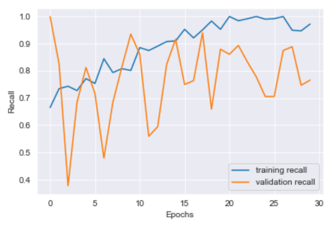
(a) Accuracy



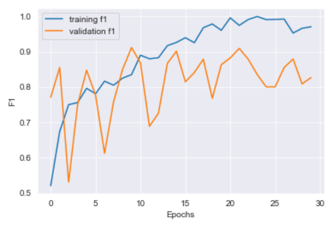
(b) Loss



(c) Precision



(d) Recall



(e) F1 score

Fig. 4: Figure with the subplots for Accuracy, Loss, Precision, Recall and F1 score through the epochs for the custom model developed.

On Table I the results of the confusion matrix and the metrics evaluated at the end of the model construction are presented. With a very high value of false negatives on the validation portion of the dataset.

TABLE I: Table for the custom model. On top the confusion matrix for the validation portion of the dataset and on the bottom the metrics obtained.

		Prediction	
		No	Yes
Actual Value	No	22(TN)	5(FP)
	Yes	11(FN)	38(TP)

	Train	Val
Loss	0.038000	0.739400
Accuracy	0.988700	0.789500
F1	0.988900	0.827100
Precision	0.988900	0.912300
Recall	0.988900	0.765900

A. *Fine-tuning*

After the model compilation and tuning and the results analysed a second try to compile on top of it, with a shorter step to try and achieve better results, was tried to avoid having to go back to the board to redesign the model.

Leading to the continuation values that are displayed on Fig. 5 as a continuation. for the previous method to try and subtle out the end part of the training process. Not resulting in an improvement for the validation data nor the training portion, an indicator of overfitting.

On Table II the results obtained after this process are presented revealling no difference than before, only decreasing the value of loss in the validation data just marginaly.

TABLE II: Table for the custom model after the second compile. On top the confusion matrix for the validation portion of the dataset and on the bottom the metrics obtained.

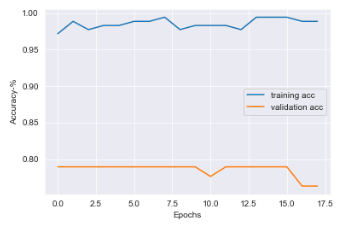
		Prediction	
		No	Yes
Actual Value	No	22(TN)	5(FP)
	Yes	11(FN)	38(TP)

	Train	Val
Loss	0.028200	0.726700
Accuracy	0.988700	0.789500
F1	0.988900	0.827100
Precision	0.988900	0.912300
Recall	0.988900	0.765900

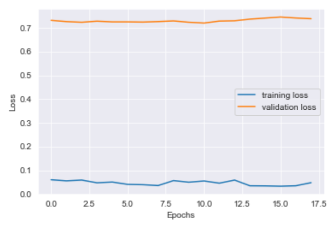
The fine-tuning process was not effective, may it be due to the limitation of the model itself or due to the implementation procedure that did not allow it to thrive by stopping the process if after 10 iterations it did not show a better accuracy that was already very high on the training data. With multiple parameters being possible of change, in particular the image size of the reading that could be increased as it is common in other models.

V. Transfer Learning

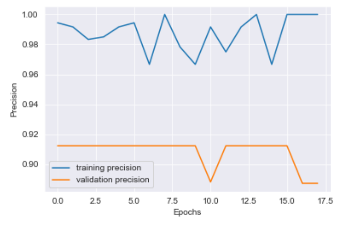
As there exist more complex methods that have already been pretrained on bigger datasets already included on the tools previously used this technique was further explored. This method is commonly used as it leads to a better performance of the



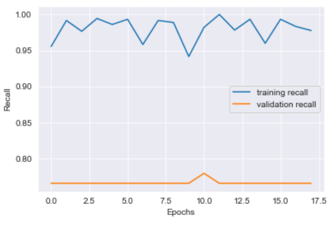
(a) Accuracy



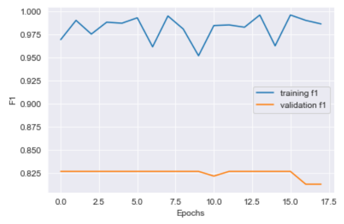
(b) Loss



(c) Precision



(d) Recall



(e) F1 score

Fig. 5: Figure with the subplots for Accuracy, Loss, Precision, Recall and F1 score through the epochs for the custom model after the second compile.

models used, being faster to develop with less computational resources being needed on the user end.

In this work the transfer learning methods were used by previously being trained on the *ImageNet* dataset, a very large dataset with more than 14,187,122 images for large-scale object recognition [18].

A. VGG19

[3] One convolutional neural network that was studied was VGG19. Created by the Visual Geometry Group at Oxford. Its architecture can be visualized in Fig. 6 where 19 layers are presented, 16 convolution layers, 3 fully connected, 5 MaxPool and 1 SoftMax [19].

To use this method the pictures were first read in RGB format and in the shape of 224 by 224 pixels. With the same transformation procedure mentioned prior to make it possible to compare methods.

To the VGG19 model with the *ImageNet* weights it was added a `Flatten` layer followed by a `Dense` one with the *sigmoid* activation function. Compiling using *binary_crossentropy* as `loss` with *Adam* with a step of 0.001

[3]Code available in `vgg19_diverse_metrics.ipynb`

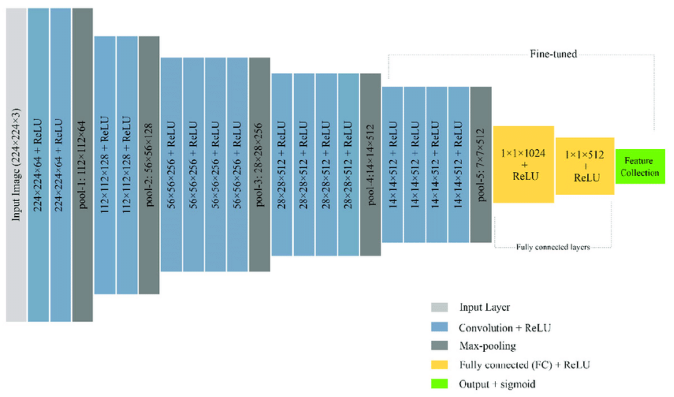


Fig. 6: Representation of the architecture behind VGG19 from [20]

as the `optimizer` and reading the metrics of accuracy, f1 score, precision and recall. Taking in consideration the `class_weight` of each class to balance the classes. Resulting in 20,049,473 total parameters with 25,089 being trainable.

When fitting the model an early stopping monitor was used, monitoring either accuracy or the f1 score. Which would stop if after 5 epochs the metric being monitored wouldn't improve, stopping earlier than the 45 epochs on a batch size each of 6 in both occasions. Taking in consideration the class weights of both classes as to balance them.

In Table III it is possible to compare the metrics obtained by the monitoring procedure of each one. Being clear that by monitoring the f1 score a better result in all the metrics is achieved.

TABLE III: Table with the metrics obtained for VGG19 by monitoring the accuracy or the f1 score on both the train portion of the dataset and the one left out for validation.

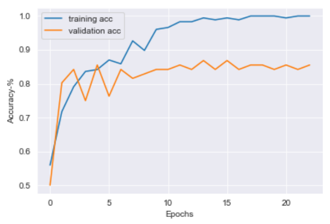
VGG19	Monitoring			
	Accuracy		F1	
	Train	Val	Train	Val
Loss	0.079956	0.322789	0.067914	0.324886
Accuracy	0.994350	0.828947	1.000000	0.855263
F1	0.994253	0.867877	1.000000	0.882039
Precision	1.000000	0.893519	1.000000	0.923611
Recall	0.988889	0.845900	1.000000	0.845900

On Table IV the results of the confusion matrix for both situations are presented. Confirmating Table III where the f1 score monitoring presents itself as being better, with 5 incorrect predictions compared to the 13 incorrect ones made if accuracy was being monitored.

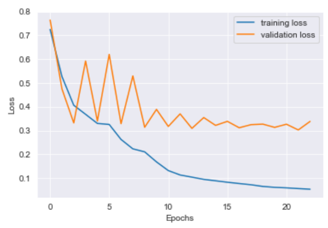
TABLE IV: Confusion matrix with the use of VGG19 on the validation part of the dataset.

VGG19		Prediction			
		Accuracy		F1	
		No	Yes	No	Yes
Actual value	No	21(TN)	6(FP)	25(TN)	2(FP)
	Yes	7(FN)	42(TP)	3(FN)	46(TP)

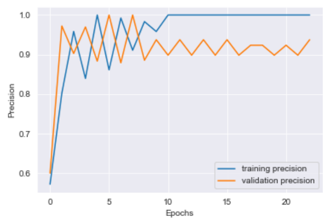
From Fig. 7 it is possible to verify how the different metrics and the loss value develop through the epochs until the value of the f1 score stabilized. With the value of loss decreasing as expected in both the training and validation sets, showing that overfitting isn't occurring. Not having the other metrics a very subtle development.



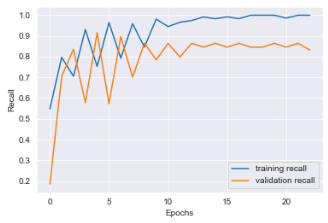
(a) Accuracy



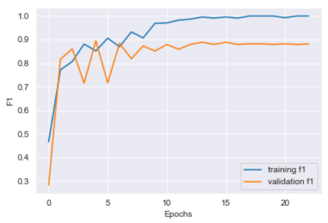
(b) Loss



(c) Precision



(d) Recall



(e) F1 score

Fig. 7: Figure with the subplots for Accuracy, Loss, Precision, Recall and F1 score through the epochs for the f1 score monitorization on both the training data and the left out for validation for VGG19.

1) Hyperparameter search: [4]

As the results obtained were better than the previous method further exploration by searching a set of parameters that could lead to a better result namely the `loss` and the `optimizer` was done. The `optimizer` from *SGD*, *RMSprop*, *Adagrad*, *Adadelta*, *Adam*, *Adamax* and *Nadam* with all the possible combinations for the `loss` of *binary_crossentropy*, *poisson* and *kl_divergence*, with cross validation on 3 folds. Maintaining the early stop method of above, studied for both the monitorization of accuracy and f1 score. On 50 epochs with a batch size of 10, not preparing the data through `ImageDataGenerator` because of the cross-validation on the 70% portion of the dataset.

Being the combination of *Adam* as the best parameter for `optimizer` and *binary_crossentropy* as the best for `loss`, with an accuracy across the cross validation of 0.830508 ± 0.013839 while monitoring the f1 score. Having this procedure failed on some fits. As did the one while monitoring accuracy.

Proving that the choice of parameters presented without testing other options was by luck the best. Leading to worse results on the validation portion of the dataset than before due to not applying data augmentation through `ImageDataGenerator`. As it is possible to visualize in the confusion matrix presented in Table V with a worse number of wrong classifications.

TABLE V: Confusion matrix with the use of VGG19 after hyperparameter search on the validation part of the dataset left out of cross validation for both occasions of accuracy and f1 monitorization.

VGG19		Prediction			
		Accuracy		F1	
		No	Yes	No	Yes
Actual value	No	21(TN)	6(FP)	23(TN)	4(FP)
	Yes	6(FN)	43(TP)	8(FN)	41(TP)

As a whole the results obtained through the method of VGG19 can be considered good as the metrics reach almost 0.90, with a better perfomance than the custom model.

B. Inception V3

[5] Another example is InceptionV3, a CNN specially conceived for the task of image analysis and object detection. Starting its life from GoogLeNet as a deep convolutional architecture in mind for constraints on memory and computational budget due to the fewer trainable parameters, 5 million, compared to the 60 million of AlexNet, one of the most famous CNNs [21]. Its architecture is the one presented in Fig. 8. Consisting on a series of 42 layers with smaller convolutions due to factorization, asymmetric convolutions, auxiliary classifiers for regularization leading to a more efficient grid size reduction and combat the vanishing gradient problem [22].

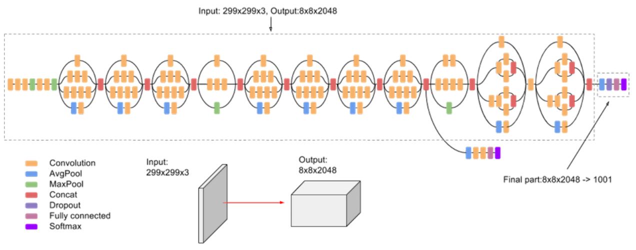


Fig. 8: Representation of the architecture behind Inception V3 from [23].

To work with this model the images needed to be read in the shape of 299 by 299 pixels and in the three channels colors, RGB. They were then normalized, divided and passed through the same `ImageDataGenerator` as before. By importing the base model of the InceptionV3 with the weights of *ImageNet* and adding at the end a `Flatten` layer and then

[4]Code available in `vgg19_hyper_parameters.ipynb`

[5]Code available in `inceptionv3_diverse_metrics.ipynb`

a `Dense` one with the *sigmoid* activation function. Resulting in a model with a total of 21,933,857 parameters with 131,073 being trainable.

To compile the model the parameters of `loss` was chosen as *binary_crossentropy* and the `optimizer` as *Adam* with a step of 0.001. With the same early stop monitor technique used for VGG19 and studying both the situations of monitoring either the accuracy or the f1 score.

In Table VI the different metrics for both situations is presented with accuracy leading to better results in the validation set and with both with better results than VGG19.

TABLE VI: Table with the metrics obtained for Inception V3 by monitoring the accuracy or the f1 score on both the train portion of the dataset and the one left out for validation.

InceptionV3	Monitoring			
	Accuracy		F1	
	Train	Val	Train	Val
Loss	0.000243	0.604363	0.001483	0.787448
Accuracy	1.000000	0.934211	1.000000	0.934211
F1	1.000000	0.953982	1.000000	0.953692
Precision	1.000000	0.965833	1.000000	0.979167
Recall	1.000000	0.944444	1.000000	0.930556

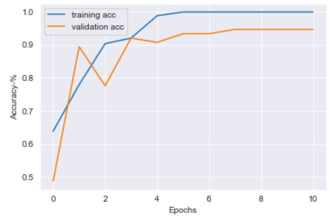
From the confusion matrix presented in Table VII the number of incorrect predictions is equal in both situations, 5 wrongly classified pictures. Less than with the VGG19 method.

TABLE VII: Confusion matrix for the Inception V3 results obtained.

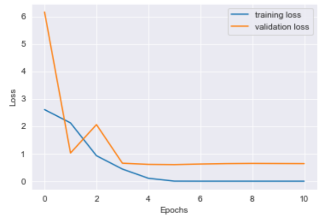
InceptionV3		Prediction			
		Accuracy		F1	
		No	Yes	No	Yes
Actual value	No	25(TN)	2(FP)	26(TN)	1(FP)
	Yes	3(FN)	46(TP)	4(FN)	45(TP)

1) Hyperparameter search: [6] As this presented itself as the best method yet further development was done. Namely the hyperparemeter search, in terms of `optimizer` from the list of *SGD*, *RMSprop*, *Adagrad*, *Adadelta*, *Adam*, *Adamax* and *Nadam* with all the possible combinations for the `loss` of *binary_crossentropy*, *poisson* and *kl_divergence*, with cross validation on 3 folds. Replicating the early stop method of above for both the monitorization of accuracy since it performed better on this model. On 50 epochs with a batch size of 10, not preparing the data through `ImageDataGenerator` because of cross-validation. Therefore, the same hyperparameter procedure as it was done in VGG19 but just by monitoring the f1 score.

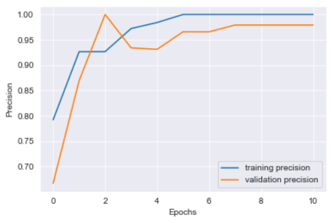
With the best choice of parameters being *binary_crossentropy* as `loss` and *Adamax* as the best `optimizer` with an accuracy score on the cross validation of 0.853107 ± 0.021139. With some other combinations having failed in their procedure. Leading to 6 incorrect predictions, 5 false negatives and 1 false positive almost the same as previously presented without this parameter search.



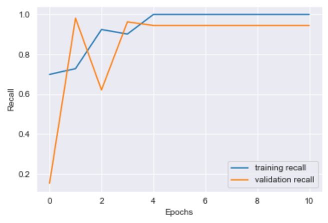
(a) Accuracy



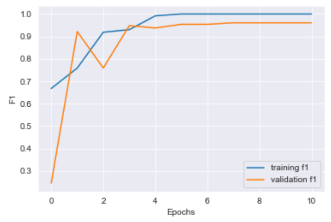
(b) Loss



(c) Precision



(d) Recall



(e) F1 score

Fig. 9: Figure with the subplots for Accuracy, Loss, Precision, Recall and F1 score through the epochs for the Accuracy monitorization on both the training data and the left out for validation for Inception V3.

The numeric differences can then be explained by the less than desirable examples where the method was tested.

2) Smaller Train set: [7] Since this method proved to lead to such good results it was chosen to see how it could handle being trained on a smaller training dataset. By using the best `loss` and `optimizer` from the previous point on just half the dataset, 126 images, meaning just using 80 tumorous pictures and 46 non tumorous for training. The half split was chosen because of splitting the data exactly in half.

With its results being presented in Table VIII. Where the values on the training set are obviously overfitting and the validation results are lower than the ones obtained on a bigger training set as it would be expected. Resulting yet on a better result than the custom method presented at the beginning.

On top of that the computational power needed was much less on the user end by using the methods of transfer learning. With Inception V3 being faster too and not requiring a lot of pictures as part of its retraining process to achieve successful results.

[6]Code available in `inceptionv3_hyper_parameters.ipynb`

[7]Code available in `inceptionv3_smaller_training.ipynb`

TABLE VIII: Table for the Inception V3 trained on the smaller set the results. On top, the confusion matrix for the test set and on the bottom the different results for the metrics.

		Prediction	
		No	Yes
Actual	No	44(TN)	8(FP)
Value	Yes	4(FN)	71(TP)

	Train	Val
Loss	0.000142	0.811070
Accuracy	1.000000	0.905512
F1	1.000000	0.916611
Precision	1.000000	0.897937
Recall	1.000000	0.940972

VI. Related work

Other works on this same topic have been developed and in specific on this dataset as well. Affirming to achieve almost 100% of accuracy on this classification task. However they mostly look at the metric of accuracy, not dealing with the fact of the unbalanced dataset. Accompanied by a very small testing portion not able to really validate the models real performance for an application in the real world.

In [24] the author makes use of the same technique for transfer learning on the VGG16 CNN and by dividing the set in 25% for testing reaches an accuracy of less than 0.85, not accounting for the unbalanced dataset by looking at other metrics to deal with the situation nor providing a confusion matrix for the final result.

Another project, developed by [25], by using the VGG16 CNN, takes the data import process a step further and crops the images to just the skull and brain portion removing unwanted dead space. Despite it producing better results it does that at the cost of reduced practicability in the real world as more computational processing is required. As a final result the value of 1.0 for accuracy was achieved by using 10 unseen pictures in the test set. Whereas in the validation portion of the test the accuracy reached 0.90. In Fig. 10 the author presented the accuracy and the loss values obtained that are worse by comparing to the work presented here. With the confusion matrices in Fig. 11 that also represent 5 wrong classifications.

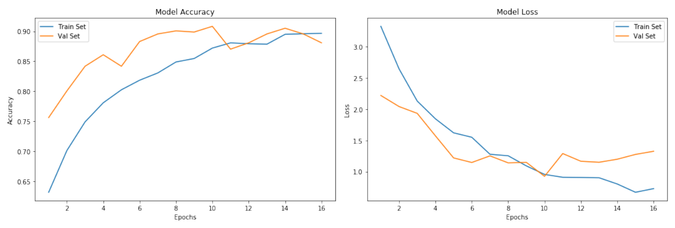
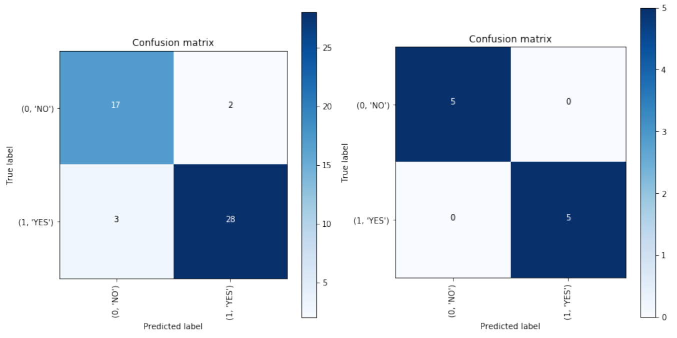


Fig. 10: Accuracy and loss for the train and validation portions of the dataset through the epochs presented in [25].

Another author studied how Convolutional eXtreme Gradient Boosting works on this dataset joined to a much larger dataset for tumor identification, meningioma or pituitary with



(a) Validation (b) Test

Fig. 11: Confusion matrices for both the validation and test portions of the dataset, on the left and on the right respectively, by [25].

a small test portion, 10%, with an accuracy in the multiclass of 0.99 [26].

Other people took the task one step further by creating real applications for its use. Have it be through the creation of training a model an integrating it on an online platform as in [27] or by doing the same for a mobile application [28]. With both leading the path for the use of techniques for the general public and as a way of helping health care providers in better and more easily doing their jobs.

Through the analysis of other works it was evident that the most common metric looked at was accuracy without carefully consideration for other metrics. With very good results being achieved in general making it clear that there is a place for deep learning techniques in the medical field as advisors.

VII. Model Comparison

By comparing the results obtained through the presented work, Tables I, II, III and IV, VI and VII and VIII it is possible to verify that the best model was Inception V3 with higher results and the best confusion matrix with 5 wrongly classified results.

The worse result was obtained by the use of the custom method showing the importance that transfer learning can have and result in a model that can be used with confidence. The second compilation and training lead to the same result with a lower loss value but not making its use of a clear importance.

The hyperparameter search for both of the methods proved itself as being ineffective for this set of parameters that were searched. Having more important ones been left out as the computational effort required would be very high to try, for instance, the learning rate. Confirming however how beneficial the process of data augmentation can be, leading to a better result as it was mentioned.

VIII. Conclusion

From this research it was verified that models with more trainable parameters as it was used in `Custom Method` may not lead to better results as it was presented by the `Transfer Learning` methods used that use much more parameters with much larger amount of data behind them which was not available here. Also verifying the importance of data balance and how some metrics can be misleading as it was done in other works. With the use of data balancing methods not have been on the reach of this work.

By analysing the problem in the general it is clear that the situation to mitigate is the number of false negatives as it could lead to less attention to the situation and result in a future situation for concern. The number of false positives are of course unwanted as they lead to the waste of medical resources that could be better spent on other patients or situations requiring more attention.

A. *Future Work*

To improve medical care provision machine learning can help make suggestions and advise to take actions earlier also making it more easily accessible to all and in a timely manner. The inclusion of images with contrast would also be an interesting add to the study. To see how the methods would deal and if better classify the presence of tumors specially the early ones. With the goal to reduce the number of false negatives. Even detecting the smallest of abnormal spots, out of human capability and warn as a future possible tumor and how this could grow over time.

On the matter of this work an interesting and important course of action would be to develop a way to scan in 3 dimensions, increasing significantly the computer power needed and in combination with object detection to determine the limits of the tumor for better local action have it be from radiation therapy, chemotherapy or surgery.

REFERENCES

[1] Institute, N. NCI Dictionary of Cancer terms. *National Cancer Institute*. (2022), https://www.cancer.gov/publications/dictionaries/cancer-terms/def/tumor

[2] McFaline-Figueroa, J. & Lee, E. Brain Tumors. *The American Journal Of Medicine*. **131**, 874-882 (2018), https://www.sciencedirect.com/science/article/pii/S0002934318300317

[3] Giovagnoli, A., Tamburini, M. & Boiardi, A. Quality of life in brain tumor patients. *Journal Of Neuro-Oncology*. **30**, 71-80 (1996,10,1), https://doi.org/10.1007/BF00177445

[4] Mabray, M., Barajas, R. & Cha, S. Modern Brain Tumor Imaging. *Btrt*. **3**, 8-23 (2015,4), http://dx.doi.org/10.14791/btrt.2015.3.1.8, doi: 10.14791/btrt.2015.3.1.8

[5] Berger A. (2002). Magnetic resonance imaging. *BMJ (Clinical research ed.)*, 324(7328), 35. https://doi.org/10.1136/bmj.324.7328.35

[6] Developer, E. What is an MRI with contrast?. *Envision Radiology*. (2022,8), https://www.envrad.com/what-is-an-mri-with-contrast/

[7] Khan, A. & Al-Habsi, S. Machine Learning in Computer Vision. *Procedia Computer Science*. **167** pp. 1444-1451 (2020), https://www.sciencedirect.com/science/article/pii/S1877050920308218, International Conference on Computational Intelligence and Data Science

[8] Khan, A., Laghari, A. & Awan, S. Machine Learning in Computer Vision: A Review. *EAI Endorsed Transactions On Scalable Information Systems*. **8** (2021,4)

[9] Deo, R. Machine Learning in Medicine. *Circulation*. **132**, 1920-1930 (2015,11,17), https://doi.org/10.1161/CIRCULATIONAHA.115.001593, doi: 10.1161/CIRCULATIONAHA.115.001593

[10] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P. & Larochelle, H. Brain tumor segmentation with Deep Neural Networks. *Medical Image Analysis*. **35** pp. 18-31 (2017), https://www.sciencedirect.com/science/article/pii/S1361841516300330

[11] Albawi, S., Mohammed, T. & Al-Zawi, S. Understanding of a convolutional neural network. *2017 International Conference On Engineering And Technology (ICET)*. pp. 1-6 (2017,8)

[12] O'Shea, K. & Nash, R. An Introduction to Convolutional Neural Networks. (arXiv,2015), https://arxiv.org/abs/1511.08458

[13] Yamashita, R., Nishio, M., Do, R. & Togashi, K. Convolutional neural networks: an overview and application in radiology. *Insights Into Imaging*. **9**, 611-629 (2018,8,1), https://doi.org/10.1007/s13244-018-0639-9

[14] Fathi, E. & Maleki Shoja, B. Chapter 9 - Deep Neural Networks for Natural Language Processing. *Computational Analysis And Understanding Of Natural Languages: Principles, Methods And Applications*. **38** pp. 229-316 (2018), https://www.sciencedirect.com/science/article/pii/S016971611830021X

[15] Brain MRI Images for Brain Tumor Detection,https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection, 10-20-2022

[16] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal Of Software Tools*. (2000)

[17] TensorFlow Tf.keras.preprocessing.image.imagedatagenerator &nbsp;: &nbsp; tensorflow V2.10.0. *TensorFlow*., https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

[18] ImageNet undefined. *ImageNet*., https://www.image-net.org/index.php

[19] Kaushik, A. Understanding the VGG19 architecture. *OpenGenus IQ: Computing Expertise &amp; Legacy*. (2020,2), https://iq.opengenus.org/vgg19-architecture/

[20] Mostafiz, R., Rahman, M., Islam, A. & Belkasim, S. machine learning & knowledge extraction Focal Liver Lesion Detection in Ultrasound Image Using Deep Feature Fusions and Super Resolution. *Machine Learning And Knowledge Extraction*. **2** (2020,7)

[21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the Inception Architecture for Computer Vision. (arXiv,2015), https://arxiv.org/abs/1512.00567

[22] T, A. Inception V3 model architecture. *OpenGenus IQ: Computing Expertise &amp; Legacy*. (2021,10), https://iq.opengenus.org/inception-v3-model-architecture/

[23] Paperswithcode Papers with code - inception-V3 explained. *Explained — Papers With Code*., https://paperswithcode.com/method/inception-v3

[24] Sailikhitarage Brain tumor detection using VGG16. *Kaggle*. (2022,9), https://www.kaggle.com/code/sailikhitarage/brain-tumor-detection-using-vgg16

[25] Ruslankl Brain tumor detection v1.0 —— CNN, VGG-16. *Kaggle*. (2019,8), https://www.kaggle.com/code/ruslankl/brain-tumor-detection-v1-0-cnn-vgg-16#5.-Conclusions

[26] Kaledhoshme123 Kaledhoshme123/convolutional-extreme-gradient-boosting-brain-tumor: Convolutional extreme gradient boosting for classification of brain tumors based on convolutional neural networks and XGBoost.. *GitHub*., https://github.com/kaledhoshme123/Convolutional-eXtreme-Gradient-Boosting-Brain-Tumor

[27] Aaliyahfiala42 Aaliyahfiala42/Data515-brain-scan-classification: This repository covers a brain scan tumor classification project for the University of Washington Data 515 course. in our project we train a CNN to predict if a MRI scan (.JPG, .PNG, .JPEG) is tumorous or not.. *GitHub*., https://github.com/aaliyahfiala42/DATA515-Brain-Scan-Classification

[28] Hoturam Brain MRI images for Brain tumor detection - 95%+. *Kaggle*. (2022,6), https://www.kaggle.com/code/hoturam/brain-mri-images-for-brain-tumor-detection-95