# Project 3 Interpretable Medical Image Classification

Anton Schäfer
ETH Zurich
Switzerland
scanton@ethz.ch

Lasse F. Wolff Anthony
ETH Zurich
Switzerland
laanthony@ethz.ch

Nidhi Agrawal
ETH Zurich
Switzerland
agrawaln@ethz.ch

## ABSTRACT

Brain tumor is an abnormal mass of tissue that forms because of unusual and uncontrolled cell growth in brain, which may lead to cancer. In order to avoid a fatal situation and provide the patient with necessary treatment, brain tumors must be detected as early and as accurately as possible. Magnetic Resonance Imaging (MRI) is a non-invasive imaging technology that produces three dimensional detailed anatomical images. Quantitative features from MRI images are then extracted by computational methods called radiomics. In this project work, we apply interpretable and explainable Machine Learning and Deep Learning methods to MRI images and radiomics dataset for quicker and more accurate brain tumor detection which can help doctors make quick decisions and supplement clinical decision making. We further discuss the interpretability and explainability of our models so as to explain how our models are making decisions. As our final classifier, we choose a simple CNN model combined with SHAP and LIME post-hoc explanations that significantly improves performance upon more interpretable and classical ML models.

## 1 INTRODUCTION

Brain tumor is one of the leading death causing diseases. Several techniques are used to diagnose brain tumor such as computerized tomography (CT) scan, electroencephalogram (EEG), but magnetic resource imaging (MRI) is the most effective and widely used method. In an MRI, magnetic fields and radio waves are utilized to generate internal images of the organs within the body. Since MRI provides more detailed information on the internal organs, it is more effective than other techniques such as CT or EEG.

Given the complex nature of tumors such as abnormalities in their sizes and their location, it is very difficult to completely understand them. In the absence of a skillful doctor such as a professional neurosurgeon, it is quite challenging to analyze the MRI reports. A radiologist generates reports from MRI manually which can be prone to errors and is quite time-consuming. Therefore, an automated system can be very helpful in clinical assistance.

In this work, we exploit machine learning and deep learning algorithms and their capabilities to recognize complex data patterns, extract features automatically, and capture long range dependencies to detect tumor in given MRI and radiomics dataset. It is extremely important to accurately interpret and explain a model's predictions. It helps to gain insight how a model can be improved and supports understanding of the process being modeled, thereby enhancing user trust. Therefore, we evaluate as well as interpret the performance of our ML and DL models and compare their interpretability and explainability.

*Machine Learning for Healthcare, 2022, ETH Zurich, Switzerland*
May 17, 2022.

## 2 MODELS AND METHODS

### 2.1 Dataset

We evaluate our models on an MRI imaging dataset and a dataset consisting of radiomics features extracted from these images:

**MRI images dataset:** The dataset consists images of 278 brain slices, 111 with tumor and 167 without any tumor, taken from the Kaggle datasets: Brain MRI Images for Brain Tumor Detection[1] and Brain Tumor Classification (MRI)[2]. Expert knowledge is not required to see the tumors in the images as they are clearly visible. The dataset is split into 250 (~89%) train and 28 (~ 11%) test samples. The images are labeled according to existence of tumor: 0 for no tumor and 1 for tumor.

**Radiomics dataset:** This dataset has been automatically generated by extraction of radiomics features such as first-order statistics, texture, and shape from the aforementioned MRI images dataset using an open-source python package called pyradiomics[3]. The split and class labels are identical.

### 2.2 Models

We evaluate several interpretable models and post-hoc explanation methods with varying complexity to investigate the trade-off between model complexity, performance, and interpretability and explainability. In the following we outline the models and methods that we use:

**Random Forest:** We employ a simple random forest classifier [3] using the radiomics features as a baseline that prioritizes performance over interpretability (Random Forest). We add a permutation based post-hoc explanation method [3] to aid in visualizing which features were the most important during classification.[4].

**Decision Tree:** A decision tree is arguable one of the most interpretable traditional ML models. The model and its training algorithm are easily understood, and we are able to interpret the model by visualizing the splits in each node. We introduce a decision tree model also trained on the radiomics features (Decision Tree) in contrast to the more complex Random Forest classifier and to highlight any potential performance cost of having a more interpretable tree-based model.

**Logistic Regression:** Additionally, we train a simple logistic regression model (Logistic Regression) that lends itself to interpretation. We obtain a clear view of the important features used the model through its linear coefficients and their magnitude. This also allows

[1] https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

[2] https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

[3] https://pyradiomics.readthedocs.io/en/latest/

[4] See https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html.

us to see which features strongly predict or contribute to the tumor or no tumor prediction by their sign and magnitude. To further aid interpretability, we use L1 regularization to force coefficients of unhelpful features to 0.

**Baseline CNN:** We train the provided baseline CNN model for increased accuracy over the simpler models (Baseline CNN). The trade-off we highlight here is model complexity, often leading to better performance, and interpretability. We add two post-hoc explanation methods, SHAP and LIME as described in the following section, to give a better understanding of the model and its predictions.

**VGG16 TL:** Finally, we evaluate a VGG16 model with transfer learning (VGG16 TL). The model is pre-trained on the ImageNet dataset. We add three dense layers followed by a dropout layer and finally the output layer. All layers in the original model are frozen and then we fine-tune our added layers on the MRI image dataset. We further employ data augmentation to add some modifications to some of our input images by making minor changes, such as flipping, enhancing their contrast and sharpness leading to increased performance.

## 2.3 Post-hoc Explanation Methods

**SHAP:** SHapley Additive exPlanations (SHAP) [5] is a model-agnostic approach that breaks down a prediction to show the impact of each feature. Image classification tasks are explained by attributing scores to each pixel on a predicted image. These scores indicate how much the pixels contribute to the probability of the image being classified in that particular class. Red pixels represent positive SHAP values that contributed positively to the classification of image as a particular class and blue pixels represent negative SHAP values which contributed to not classifying the image as that class.

**LIME:** Local interpretable model-agnostic explanations (LIME) [8] is, like SHAP, a model-agnostic approach for explainability. LIME explains individual predictions of a classifier by learning an interpretable and faithful model locally around each prediction. In our task of brain tumor detection from MRI images, LIME produces images that explain our classifiers' predictions by highlighting the important region inside the image.

# 3 EXPERIMENTS

## 3.1 Experimental Setup

All of our experiments were performed on a 2021 MacBook Pro (Model Identifier MacbookPro18,3) with the Apple M1 Pro chip and 16 GB of memory running macOS Monterey 12.3.1. We repeated every classical ML model run five times with consecutive seeds (seed 42 to 46) and report the mean and standard deviation. DL models were only run once using seed 0.

The tree-based models and the logistic regression classifier were developed using Scikit-learn [7]. We used PyTorch [6] to develop the CNN models and Tensorflow [1] the VGG16 transfer learning model. The CNN models were trained with the Adam optimizer [4] using a learning rate of 0.001 and batch size of 64 for a maximum of 50 epochs using early stopping. Convergence was assumed when the cross-entropy loss did not improve on the validation set for 5 consecutive epochs. The VGG16 transfer learning model was also trained using the Adam optimizer but for 100 epochs with an exponentially decaying learning rate of 0.001 and a batch size of 128. Again, we used early stopping but with a patience of 7 epochs. The final models are those with the lowest validation loss.

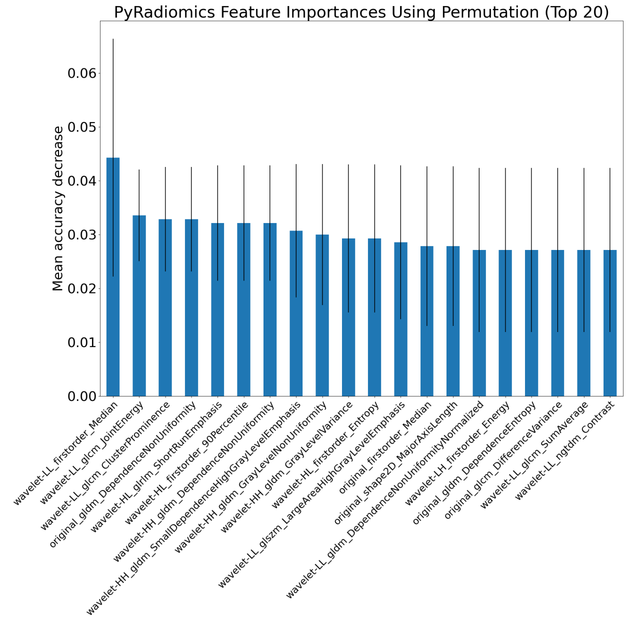


**Figure 1: Top 20 features used by our Random Forest classifier determined by a permutation-based feature importance method [3]. Each feature is permuted 50 times.**

Hyperparameters were tuned on a hold-out validation set using extensive random searches [2]. Optimal configurations can be found in the source code. Our final models were trained on the combined training and validation set to further increase performance.

## 3.2 Results and Discussion

Table 1 shows the performance of our evaluated models on the the test sets. We observe that the more complex models show significantly better performance both in terms of accuracy and F1-score compared to the interpretable and simpler models. The VGG16 transfer learning model and pre-training on ImageNet, however, performed worse than our CNN baseline, likely due to the small size of our dataset and overfitting to the training and validation data. We also notice that the random forest classifier does not seem to improve upon the more interpretable decision tree on our dataset. Again, this is likely due to the small size of the dataset and overfitting.

We add several visualizations of our models to aid interpretability and explainability. In Figure 2 we visualize the splits of each node in the decision tree to explain its predictions. This visualization further suggests which features are important in classifying whether there are a visible brain tumors or not in the MRI images. In Figure 1 we visualize the top 20 features used by our random forest classifier determined by a permutation-based feature importance method

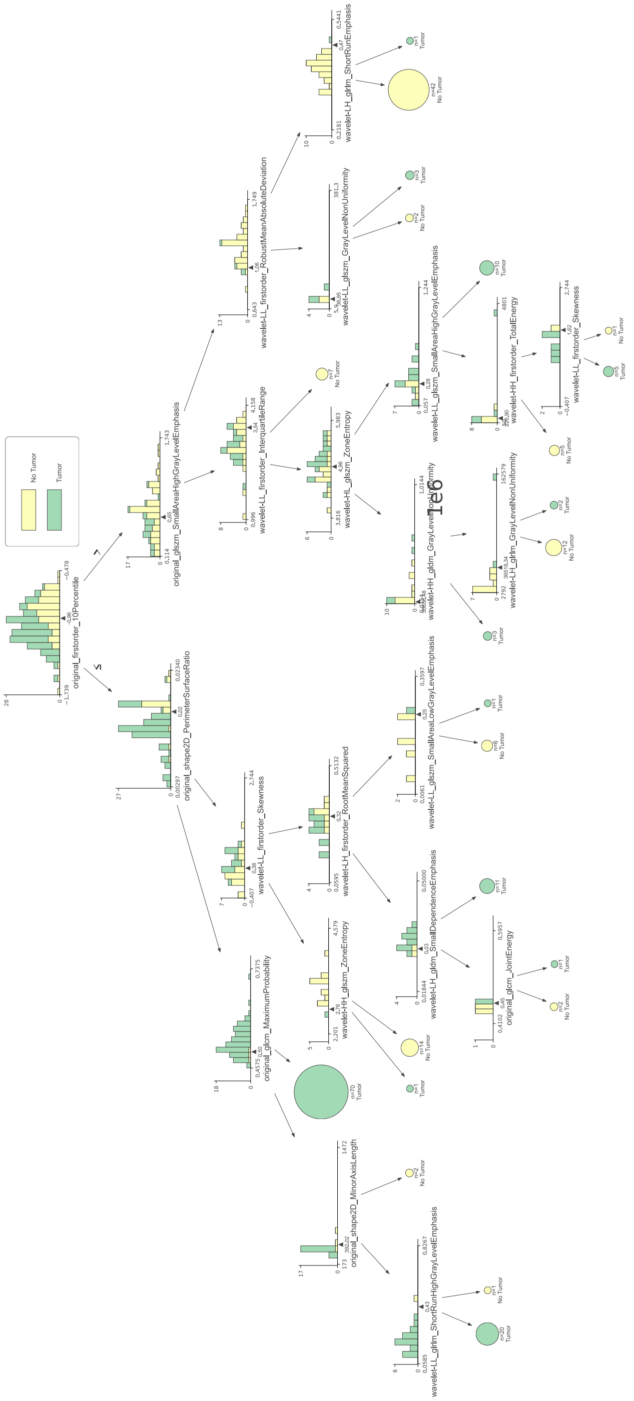

No Tumor
Tumor
original_firstorder_10Percentile
original_glszm_SmallAreaHighGrayLevelEmphasis
original_shape2D_PerimeterSurfaceRatio
wavelet-LL_firstorder_RobustMeanAbsoluteDeviation
wavelet-LL_firstorder_InterquartileRange
wavelet-LL_firstorder_Skewness
original_gldm_MaximumProbability
wavelet-LH_glrlm_ShortRunEmphasis
wavelet-LL_glszm_GrayLevelNonUniformity
wavelet-HL_glszm_ZoneEntropy
wavelet-LH_firstorder_RootMeanSquared
wavelet-HH_glszm_ZoneEntropy
original_shape2D_MinorAxisLength
wavelet-LL_glszm_SmallAreaHighGrayLevelEmphasis
wavelet-HH_firstorder_TotalEnergy
wavelet-LL_firstorder_Skewness
wavelet-LH_glrlm_GrayLevelNonUniformity
wavelet-LL_glszm_SmallAreaLowGrayLevelEmphasis
wavelet-LH_gldm_SmallDependenceEmphasis
original_glcm_JointEnergy
wavelet-LL_glrlm_ShortRunHighGrayLevelEmphasis
n=70
Tumor
n=42
No Tumor

**Figure 2: Visualization of our Decision Tree classifier. The visualization is generated using the dtreeviz library.**

[3]. First-order wavelet features seem to be the most important in its predictions. Similarly, in Figure 3 and Figure 4 we visualize the most important features of our logistic regression classifier.

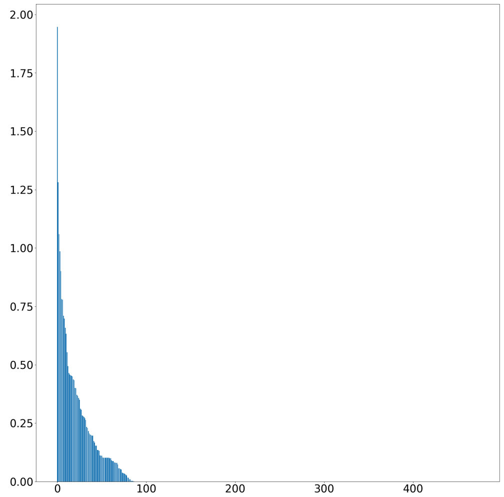


**Figure 3: Absolute coefficients of our Logistic Regression classifier sorted by magnitude.**

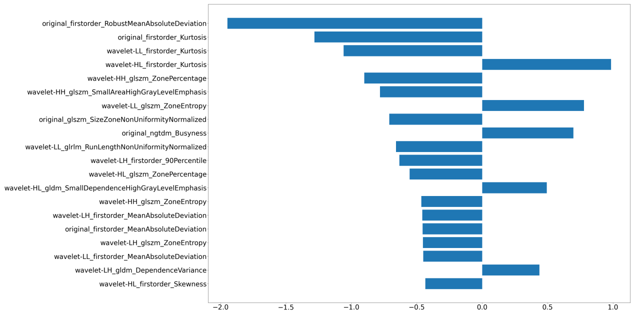
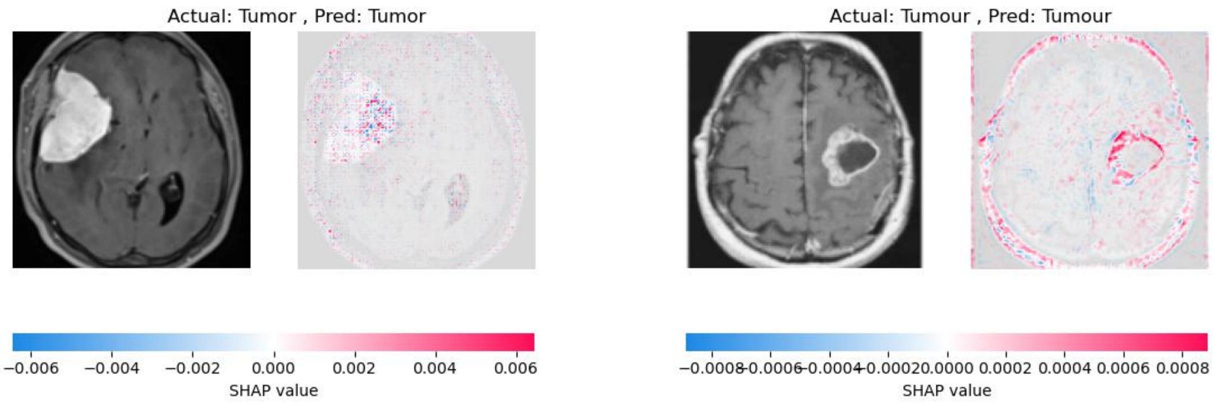


**Figure 4: Top 20 most important features of our Logistic Regression classifier and their corresponding coefficients.**

Only ∼ 100 coefficients are non-zero and again first-order wavelet features seem play an important role in its predictions.

For our DL-based models, we visualize our post-hoc explanation methods in Figure 5 and Figure 6. In Figure 5a, we can clearly see that our CNN model is focusing on the area within the brain that has a tumor according to the SHAP values and thus explaining the prediction. Similar patterns can be noticed in Figure 5b for our VGG16 transfer learning model. Misclassifications and their corresponding SHAP values can be found in Appendix A. Lastly, in Figure 6 we visualize the LIME explanations for our CNN baseline. Again, we can see that the model highlights the interesting regions, specifically we see that the entire tumor region is highlighted for the tumor prediction. The highlighted regions may be especially important in a clinical setting, where specialists are able to quickly verify the models' predictions by inspecting these regions.

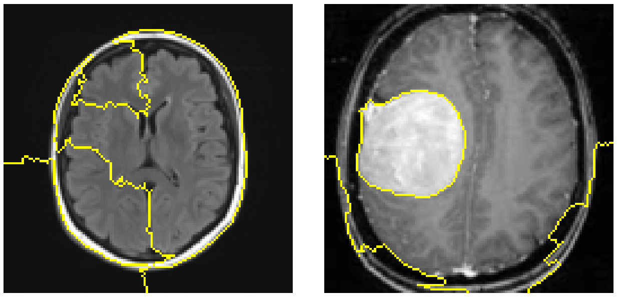
Based on our visualizations as well as our models' performance, we recommend the Baseline CNN as the optimal classifier trading off interpretability for significantly increased performance. Both SHAP and LIME provide adequate explanations of the model's



**(a)**

**(b)**

**Figure 5: SHAP images. (a) Baseline CNN (Actual: Tumor, Pred: Tumor). (b) VGG16 TL (Actual: Tumor, Pred: Tumor).**



**(a)**

**(b)**

**Figure 6: Baseline CNN LIME explanations. (a) Actual: No Tumor, Pred: No Tumor. (b) Actual: Tumor, Pred: Tumor.**

prediction and would help increase trustworthiness in a clinical setting even if CNNs are otherwise black boxes and offer very limited interpretability.

## 4 CONCLUSION

In this work, we have evaluated several interpretable models, such as decision trees and logistic regression, and post-hoc explanation methods, like convolutional neural networks and VGG16 with LIME and SHAP in the task of brain tumor detection based on magnetic resonance imaging scans. We further introduced a random forest baseline together with a permutation-based feature importance method as a more complex baseline with limited interpretability. We showed that the simpler and more interpretable models do not offer the same performance as more complex models combined with post-hoc explanation methods but the simpler models are more easily understood and interpreted. If performance is crucial and post-hoc explanations are sufficient, convolutional neural networks are a good choice with high performance and explainability with methods such as SHAP and LIME.

| Model | Train time (s) | Accuracy | F1 |
|---|---|---|---|
| Baseline CNN | 34 ±0 | **0.9286** ±0.0000 | **0.9545** ±0.0000 |
| VGG16 TL | 34 ±0 | 0.8571 ±0.0000 | 0.9000 ±0.0000 |
| Logistic Regression | 3 ±0 | 0.7857 ±0.0000 | 0.8500 ±0.0000 |
| Decision Tree | **0** ±0 | 0.7571 ±0.0143 | 0.8190 ±0.0121 |
| Random Forest | **0** ±0 | 0.7571 ±0.0267 | 0.8279 ±0.0224 |

**Table 1: Model performance on their respective test sets.**

## REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research* 13, 2 (2012).

[3] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[4] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[5] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach,

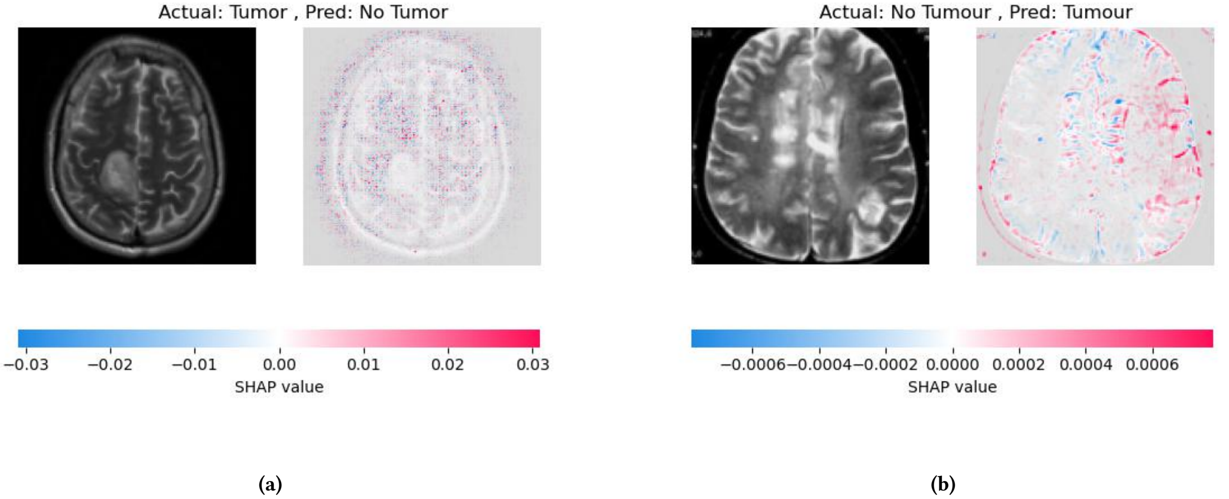
H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 1135–1144.

# A MISCLASSIFIED SHAP IMAGES

Figure 7 shows misclassifications for our CNN baseline and VGG16 transfer learning model.



**(a)**

**(b)**

**Figure 7: SHAP images. (a) Baseline CNN (Actual: Tumor, Pred: No Tumor). (b) VGG16 TL (Actual: No Tumor, Pred: Tumor).**