applied sciences

MDPI

Article

AI Approaches in Computer-Aided Diagnosis and Recognition of Neoplastic Changes in MRI Brain Images

Jakub Kluk * and Marek R. Ogiela *

Cryptography and Cognitive Informatics Laboratory, AGH University of Science and Technology, 30 Mickiewicza Ave., 30-059 Krakow, Poland

* Correspondence: klukjakub@gmail.com (J.K.); mogiela@agh.edu.pl (M.R.O.)

Abstract: Advanced diagnosis systems provide doctors with an abundance of high-quality data, which allows for diagnosing dangerous diseases, such as brain cancers. Unfortunately, humans flooded with such plentiful information might overlook tumor symptoms. Hence, diagnostical devices are becoming more commonly combined with software systems, enhancing the decisioning process. This work picks up the subject of designing a neural network based system that allows for automatic brain tumor diagnosis from MRI images and points out important areas. The application intends to speed up the diagnosis and lower the risk of slipping up on a neoplastic lesion. The study based on two types of neural networks, Convolutional Neural Networks and Vision Transformers, aimed to assess the capabilities of the innovative ViT and its possible future evolution compared with well-known CNNs. The research reveals a tumor recognition rate as high as 90% with both architectures, while the Vision Transformer turned out to be easier to train and provided more detailed decision reasoning. The results show that computer-aided diagnosis and ViTs might be a significant part of modern medicine development in IoT and healthcare systems.

Keywords: artificial neural networks; computer vision; magnetic resonance imaging; convolutional networks; vision transformer; cancerous diseases

Citation: Kluk, J.; Ogiela, M.R. AI Approaches in Computer-Aided Diagnosis and Recognition of Neoplastic Changes in MRI Brain Images. *Appl. Sci.* **2022**, *12*, 11880. https://doi.org/10.3390/app122311880

Academic Editor: Luis Javier Garcia Villalba

Received: 28 October 2022
Accepted: 19 November 2022
Published: 22 November 2022

1. Introduction

Cancerous diseases are currently one of the biggest challenges in modern medicine. They are responsible for almost a quarter of all deaths worldwide (24% in 2019) [1], which makes them the second most common cause of mortality, just after circulatory diseases. The reason for that is the comparably low detection rate in conjunction with high treatment costs and potential tumor recurrence. There are many types of neoplastic diseases, and most require specific treatment and diagnostic methods, making cancers a varied group of sicknesses. Tumors causing the most deaths include lung, colorectal, breast, or prostate cancer, accumulating up to 44% of cancer-related deaths. Listed neoplastic diseases combine commonness and lethality as reasons for such a high share, but there are also much deadlier types of tumors. An example of that might be a group of afflictions of the central nervous system, including brain tumors. While brain tumors might not be the most common group of neoplastic diseases (approximately 1.8% of cancers worldwide in 2020), they are among the deadliest, causing over 250 thousand deaths in 2020 [2], with the five-year survival rate varying from 75% for children to only 21% for middle-aged people [3]. Additionally, brain tumor is one of the most common cancerous diseases among children, responsible for 26% of all cancer cases, making them the second most frequent neoplastic after leukemia [4,5].

Regarding the WHO Classification of Tumors of the Central Nervous System [6], there are 12 categories of brain tumors out of which 120 different types are further distinguished. Finding a comprehensive dataset with sample images of all the possible classes is impossible as this data is rare, non-balanced, and restricted due to privacy and data ownership reasons [7,8]. Additionally, artificial intelligence algorithms require specific data labeling,

which needs to be done by a professional, which often is not a priority and increases costs, leading to fewer properly labeled data sources. Thus, this work focuses on a publicly available dataset [9], consisting of three tumor types: glioma, meningioma, and pituitary tumors. Gliomas account for 80% of all malignant brain tumors, some subtypes being the most common in the case of adults (Glioblastoma) and others the most common in the case of children. Meningiomas, in turn, are one of the most frequent non-malignant brain cancers among adults (over 50% of all cases), while pituitary tumors are typical non-malignant cases among children (almost 20% of all cases) [4,5].

As brain cancers show symptoms only at a late stage of development, which combined with the fact that even recognition of the tumor type might require an invasive examination, such as a biopsy, makes them a group of diseases requiring specifically robust diagnostic methods. Magnetic Resonance Imaging (MRI) is considered one of the best examination types regarding central nervous system tumors. It is an imaging method that provides doctors with plentiful information about the presence of any pathological lesions, their location, size, malignancy, and even specific types [10]. However, this information does not result directly from the examination, it has to be extracted from the pixels. As the MRI method results in dozens of grayscale pictures, it creates a lot of data that needs interpretation. Leaving this task for a doctor to complete manually not only increases the time required for a diagnosis but also affects the method's robustness, as it is possible to overlook the lesion, especially while looking at a significant number of similar images in IoT healthcare systems. That is the motivation behind Computer Aided Detection (CAD) algorithms [11].

The CAD systems help doctors by taking the input data and extracting some information; one of the techniques would be to classify examination data as sick or healthy or, for example, in the case of cancer—detect specific tumor types. This classification task can be approached in a few different ways, starting from classical machine learning, where feature extraction and selection were separated from decisioning, ending on methods based on Deep Learning [12]. Currently, in the specific application of CAD systems supporting image-based diagnosis methods, as in most image-related use cases, Convolutional Neural Networks (CNN) are state-of-the-art solutions. Especially in the case of brain tumor classification, there are works achieving accuracy varying at approximately 90% with different datasets and various types of CNNs [11,13]. The concept of this work raises the question of whether there are algorithms that may outperform convolutions.

The general idea of CAD systems is not to replace doctors but to support them, as making a diagnosis based purely on an unexplainable software model might pose a significant threat to patients. Considering that the program detecting neoplastic changes cannot simply label MRI images as cancer-containing or not, it needs to explain its decision. A natural approach to explaining in the case of image data is by marking out areas of the image that influenced the decision the most. In the case of CNNs, it can be implemented, for example, by using the Grad-CAM algorithm [14]. This procedure is not the most comprehensive, as it only marks what the network "thinks" is important, not what is truly important. The system can recognize, for example, the image's background as an indication of cancer or tag something as a tumor that seems unrelated to the disease. There also might be a problem with the interpretation of labels on healthy images, as there will not be anything specific to make the picture healthy. In the case of a tumor-free MRI shot, no areas should be marked as exceptionally interesting, but the Grad-CAM will result in such markings. This issue could be addressed with supervised-learning segmentation networks, but such a solution would require a significant amount of labeled data, which is hardly accessible in the medical world.

This work addresses the problem of the creation of a CAD system recognizing different types of brain tumors based on MRI scans along with marking areas containing pathological lesions in an unsupervised manner. The goal is to achieve the accuracy presented in related works [13,15], while training on labeled pictures and producing high-quality region-of-interest tagging without any segmentation. In addition, the research aims to assess the

future development paths of CAD systems for image-based diagnosis support by using two approaches to image classification: current state-of-the-art CNNs solutions and a novel, attention-based architecture called Vision Transformer (ViT) [16]. The goal is to compare classification outcomes, ease of training, and the quality of resulting regions of interest using both trained-from-scratch and pre-trained versions of algorithms [17].

The rest of the article is structured in the following way. Section 2 introduces and compares CNN and ViT architectures and describes methods used in both systems to acquire regions of interest. Later in this section, the data used for training and evaluation is presented. Section 3 contains both numerical and pictorial results. Section 4 discusses the results, training course, and output quality, summarizing the authors' opinions on possible future advancements in the field.

2. Materials and Methods

2.1. Convolutional Neural Networks and Grad-CAM Visualization

Convolutional Neural Networks are currently the state-of-the-art solution for most artificial intelligence tasks related to images. This architecture is a root for Deep Learning due to its ability to learn a hierarchical representation of the data, which made it possible to skip the feature extraction part—an indispensable step in classical machine learning. CNNs use a convolution kernel instead of a vector of connection weights to roll over a picture to extract local relations between pixels. Further convolutional layers do the same thing on the outputs from previous ones, extracting more globally-oriented features in each layer [12]. This approach allows for the creation of high-level image representations that the classifier can use to make decisions. Due to the large number of different convolution kernels learned by the network, provided the dataset contained suitable samples, the algorithm is also translation, scale, and rotation invariant [18].

While powerful, CNNs, as most Neural Network algorithms, lack interpretability. It is hard to tell why the network made a specific decision, which is unacceptable in the case of a CAD system. Fortunately, some algorithms allow visualizing the inner workings of a CNN. One way to visualize the network's inner workings is to synthesize an input that maximizes the layer's activation, showing what features this neuron has learned [19]; this method is called Activation Maximization and is presented on the left in Figure 1. This approach is excellent when it comes to understanding what a CNN's layer "sees", but it does not help to judge why a particular input has been marked as a specific label. When considering reasons behind a decision regarding a specific input, one of the most popular visualization methods is Grad-CAM, also presented in Figure 1 on the right.

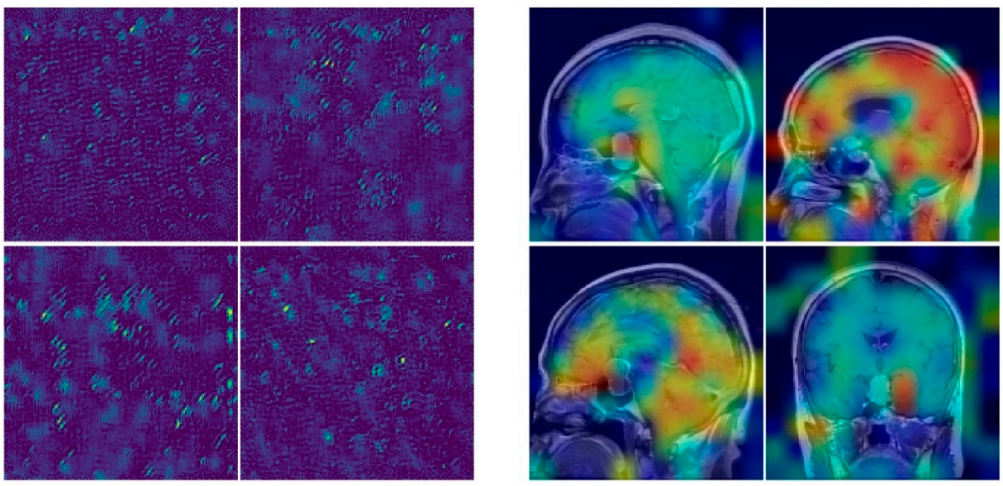

Figure 1. An Activation Maximization (on the **left**) and Grad-CAM visualization (on the **right**) examples of a CNN trained to classify brain tumors.

As seen in Figure 1, the advantage of the Grad-CAM method is that it is input-specific; it does not answer the question of "how" the neural network resolved the problem but which areas of the image influenced the response the most. An added advantage of this algorithm is that it does not have to mark areas responsible for the actual response; it can create a visualization of all the classes learned by the network. It might be instrumental in a CAD system, for example, in the case of non-tumor classification to highlight suspicious areas resembling tumors about which the network was unsure.

The name of the algorithm explains the idea behind it. Grad-CAM stands for: "Gradient Weighted Class Activation Mapping", which means that the importance of every pixel is calculated based on the gradients of the given output class relative to feature activation maps of the last convolutional layer. Every convolutional layer in the network operates on several filters, producing a single feature map (mostly a two-dimensional table). These feature maps are then flattened and used by the classifier (usually a fully-connected neural network) to calculate the response of the whole network. For every value in these feature maps, the gradient can be calculated using the weights of the classifier. These gradients are then normalized, and a weighted point-wise mean of feature maps is calculated, with the sum of gradients used as a weight. The result of these computations is a single, averaged feature map that can be treated as an image. This image then can be upscaled to the size of the input and superimposed with it, creating a heatmap, where high values of the resulting feature map indicate the great importance of the pixel and low values reveal that the pixel was not relevant for the result. In Figure 1, high pixel importance is presented using red, while non-important pixels are painted blue, and such an approach shall continue later in the work. More details about the Grad-CAM algorithm can be found here: [14].

2.2. Attention Mechanism and Vision Transformers

Despite being one of the best-known and certainly the most preferred algorithms, Convolutional Neural Networks are not the only option for computer vision. An alternative presented in this work uses the Attention Mechanism. As with many of the neural networks' features, this method was designed to mimic how the human brain works. The idea is to help the network to focus on what is vital in the input data instead of treating all the inputs equally. This mechanism differs from static weights by being dynamically computed per input, which allows it to find essential features regardless of their position in the input.

The attention to Deep Learning was first introduced in Natural Language Processing (NLP) related tasks, especially machine translation [20]. This method was used in encoder-decoder architectures based on Recurrent Neural Networks (RNN) or its variants, such as Long-Short Term Memory (LSTM), to remedy the problem of performance drops with long sentences as inputs. The attention measured how influential words from the input sentence were regarding the output sentence and did it for each pair of words. In some use cases, attention was also computed for pairs of words from a single statement, extracting the information about which words mattered the most [21].

Various forms of attention in combination with RNNs became popular in NLP, which led to architectures relying on it more than on recurrency until, at some point, an encoder-decoder model for machine translation based purely on attention was created [22]. This model was a Transformer, and it was meant to achieve translation accuracy comparable with RNNs, while eliminating their shortcomings. As recurrent networks work in the time domain by calculating their hidden states one at a time, it is hard to parallelize computations, thus making them slow. Getting rid of recurrency in favor of attention allows for taking advantage of modern hardware, such as Graphical Processing Units (GPUs) or Tensor Processing Units (TPUs).

The Transformer architecture is also an encoder-decoder model built of multiple processing block series that can be graphically represented as in Figure 2, where the "Nx" sign is the number of block repetitions. Both the encoder and decoder blocks are built of sets of operations arranged in a different order: Multi-Head Attention (masked or not), residual connection with normalization [23], and simple Multilayer Perceptrons (MLP), also called as Feed-Forward Networks [12]. The difference between a decoder and an encoder is an additional Masked Multi-Head Attention and normalization operations in the former and different inputs. The encoder's blocks accept as an input the network's input (the first block) and a previous encoder layer's output. In contrast, the decoder accepts analogous information with the addition of the encoder's corresponding layer's output.

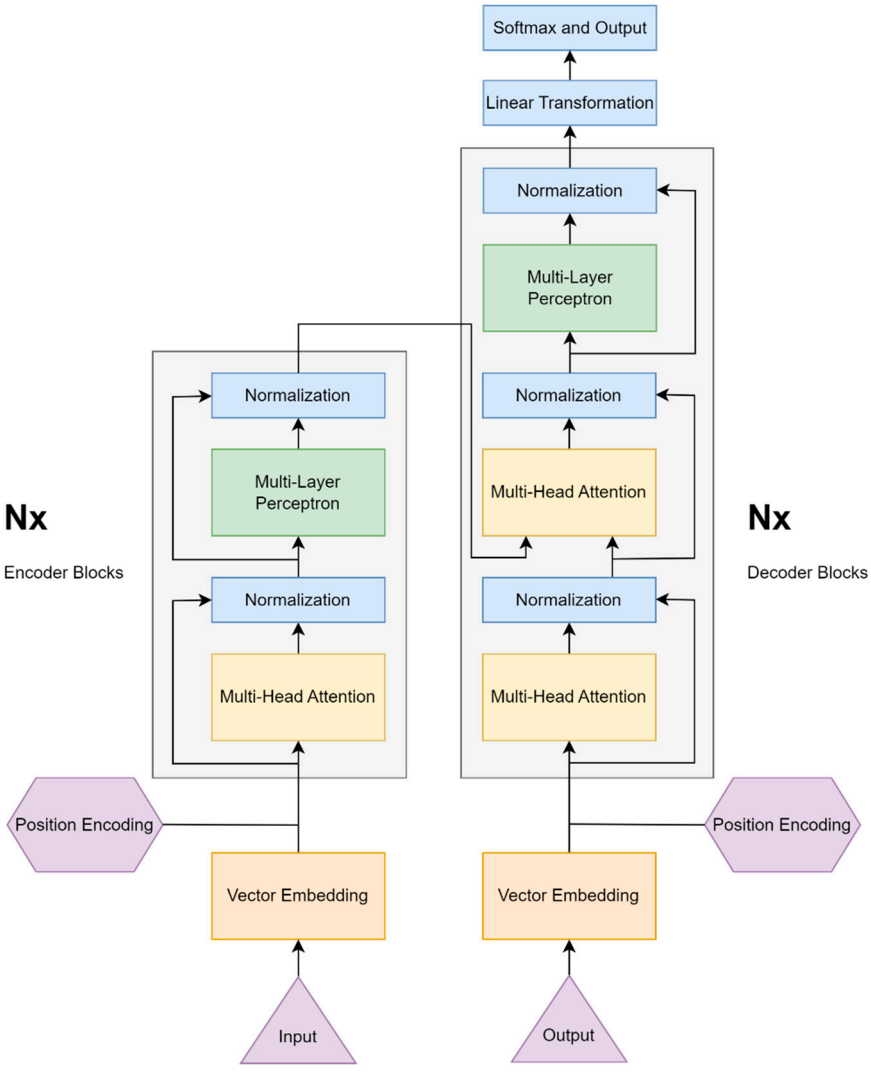

Figure 2. The Transformer architecture.

The Transformer translates the input data into some form of internal representation. Each layer's output size is a fixed value shared between every block and its internal parts. The translation happens by calculating the Multi-Head Attention of inputs and then transforming them with a simple MLP network. The attention extracts essential pieces of information, and the MLP projects them into the latent representation. Thus, the Multi-Head Attention works as a generalization for the Feed-Forward with dynamically computed weights, making it the crucial part of the model. Many different forms of attention that can be used in Deep Learning have been developed and described [24]. As this work concerns Transformers, the one described in detail is the same used by the authors of the architecture [22].

The Multi-Head Attention implemented in the Transformer can be described as a three-input block consisting of a few operations. In some cases, each input might be the same vector (as in the encoder) thus, the name self-attention. The heart of the block is the Scaled Dot-Product Attention, which multiplies input vectors (or matrices in case of stacked inputs for faster computations), scales them by the square root of inputs dimensionality, and normalizes with a softmax function, as shown in the Equation (1) [22]. This operation is not performed once for a triplet of inputs. Instead, it is done repeatedly with inputs previously multiplied by different matrices with learnable parameters that cast them into a smaller dimension. Calculating attention multiple times on reformatted data allows focusing on different parts of vectors, similar to multiple filters in a single convolutional layer of a CNN. Obtained matrices are stacked together and again cast linearly to the model's inner dimensionality. As inputs are cast to lower dimensions, these multiple computations do not slow down the whole process compared to single full-dimensionality attention. Both operations-Scaled Dot-Product Attention and Multi-Head Attention, can be visualized in the form of block diagrams in Figure 3.

$$Attention(Q,\ K,\ V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

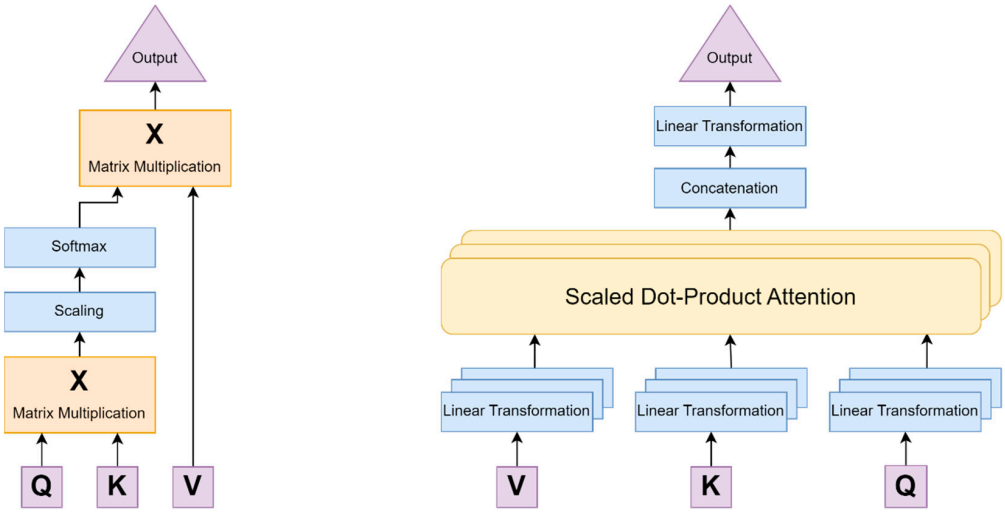

Figure 3. Scaled Dot-Product Attention (**left**) and Multi-Head Attention (**right**) graphical representations.

The described mechanism can extract global and local features from the input vectors, although the attention works globally and does not assess anything about input structure. The information about spatial relations between parts of the input does not come from the model's inductive bias, as in RNNs or CNNs, so to draw out these associations, the locality needs to be encoded into the data. In the case of the Transformer model, the sequence order is encoded into the inputs. In the NLP task, input sequences were translated into vectors using one of the well-known techniques [25], these vectors were then summed with learned position embeddings. These embeddings are vectors of the same dimensionality as inputs that indicates each element's position. There are numerous methods for obtaining such positional encoding [26]. In this case, vectors of sin/cos waves of exponentially changing frequency were used.

The architecture described above became one of the best algorithms for NLP tasks that largely replaced RNNs and LSTMs, but the attention itself is not suited explicitly for processing language. It is a general computational method valuable in a variety of different problems; it has been used in computer vision too, but mostly in conjunction with different techniques, such as CNNs, and did not provide far better results than well-established strategies [27]. The reason for that lies in calculating attention for every pair of input pieces, which means it is a quadratic operation. It was immaterial in the case of sentences, as even very long clauses contain, at most, dozens of words, while even small pictures consist

of thousands of pixels; a popular dataset, ILSVRC (a subset of ImageNet) [28], contains images of mean size of approximately 400 × 400 pixels. Trying to compute the Multi-Head Attention on such an input, in most cases, leads to a lack of memory. Some approaches tried to mitigate that problem by limiting attention pairs, for example, by calculating it only locally, for pixels in close proximity [29], which resulted in behavior very similar to convolution kernels, or by calculating it more sparsely, for only a subset of pixels [30].

The architecture used in this work, the Vision Transformer (ViT) [16], approaches the problem differently: instead of altering the attention mechanism to work with images, it adjusts the data and leaves the algorithm untouched. In the ViT architecture shown in Figure 4, the Transformer Encoder block is built identically to that used in NLP. What changes between these two models is what happens with inputs and outputs. Creation of the Transformer's input from an image requires dividing the picture into not overlapping patches of a fixed size, flattening them, which results in a set of vectors similar to sentence embeddings, and linearly projecting to ViT's latent space dimensionality via learnable matrices. The vectors prepared in this way are then combined with positional embeddings, which is done the same way as when ordering sentence elements because creating embeddings able to encode the two-dimensional location of a patch did not bring improvement.

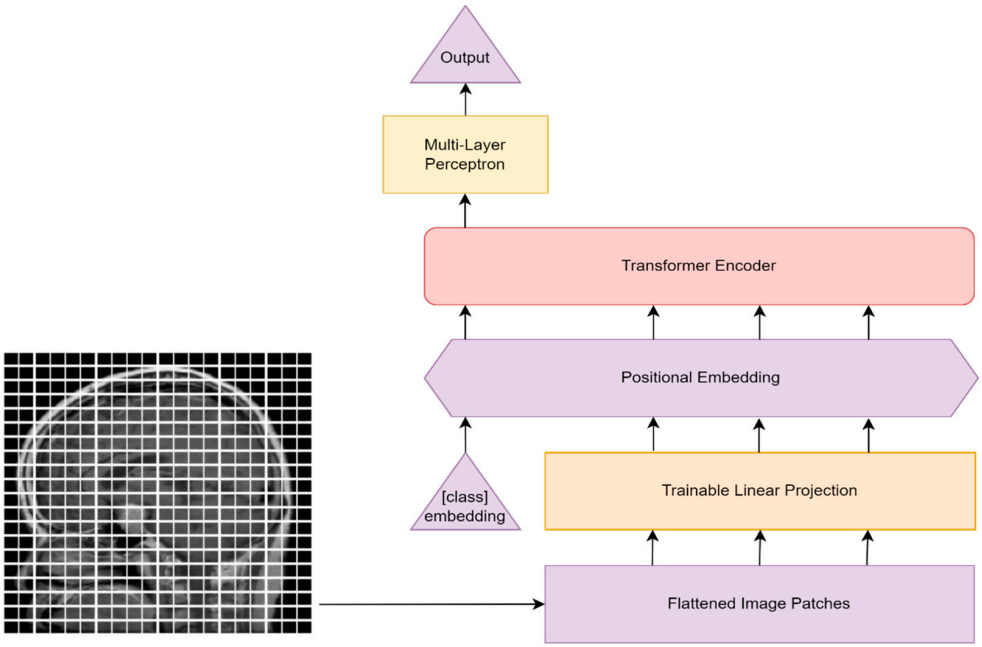

Figure 4. The Vision Transformer architecture.

There is no decoder block in ViT since the latent space vector of the model is a sufficient representation of the input and can be used for classification. The outputs of the last encoder block go into an MLP network that uses it to make a decision. However, this architecture implements a special type of input - a learnable class embedding similar to the BERT's (cls) token [31]. This token is meant to represent the picture in the form of a single vector, which allows for using it alone for decisioning without even computing further outputs from the last block. This design allows the Transformer's encoder to serve a purpose similar to convolutional layers in a typical CNN: to extract features from the image. The difference is that the ViT has a lesser inductive bias towards understanding images than a CNN, which consequently requires more data to train correctly, but with the added value of extracting global connections from the early processing stages.

It is possible to explain Vision Transformer's decision by visualizing which parts of the image influenced it the most. It can be done similarly to CNNs, using the Grad-CAM algorithm, but there is also a different way of visualizing the network's inner processes. The method is called Attention Rollout [32], and it can be used to plot the value of attention, which is what matters in the network's decisioning. The procedure is to average the attention weights in all layers and to multiply attention matrices from different layers recursively, which reflects the information flow in the network. The advantage of this method is that it is able to revert the information flow in the network and visualize points of interest with the input's resolution without upscaling needed in Grad-CAM.

2.3. Dataset Description

Obtaining a dataset of a decent size and high annotation quality in the field of medical data is a challenging task. Despite a large number of medical examinations, including MRI scans of the brain in a search for tumors conducted every year, there is a lack of publicly available datasets containing this information. As MRI scans are a type of personal data, they need to be protected, which requires compliance with the procedures while working with data and their proper anonymization, which often leads to losing valuable details. These datasets that are publicly available are often relatively small for the task of training neural networks and are often poorly annotated, thus making a reliable CAD system challenging. Due to these restrictions, this work focuses on recognizing only three types of brain tumors: pituitary, glioma, and meningioma. Of the types mentioned here, the glioma tumor turns out to be the most common malignant neoplasm, which is one of the reasons tumor type recognition is essential. The created system also takes into consideration the no-tumor case.

The dataset used for training came from the Kaggle platform and was created by Sartaj Bhuvaji et al. [9]. It consists of images coming from different internet sources that were collected, filtered, and annotation-checked by authors and then published under the Creative Commons 1.0 license. The dataset contains 3264 images from four categories: no_tumor (500), pituitary_tumor (901), glioma_tumor (926), and meningioma_tumor (937). Images come in all three different cross-sections: Sagittal, Coronal, and Transverse planes, although the dataset is not balanced regarding the type of plane visible on the image, and they are not labeled with that information. All the images are grayscale, and their resolution varies from 167×174 to 1446×1375 pixels, but the most common resolution is approximately 500×500 pixels.

Before training networks, all the images were processed with a set of operations: rescaling to the size of 160×160 pixels, normalization to the (0, 1) interval, and addition of one-hot encoding annotation; every image was paired with a four-element vector of zeros and a single one, indicating from which class the image comes. The dataset was then split into training and testing in a proportion of 70%:30%, and the training dataset was subjected to data augmentation with operations: random rotation in the range of $(-15,15)$ degrees, vertical and horizontal translation up to 10% of image's size, and vertical and horizontal flips. The test dataset was not augmented. A few random samples from the training dataset are shown in Figure 5.

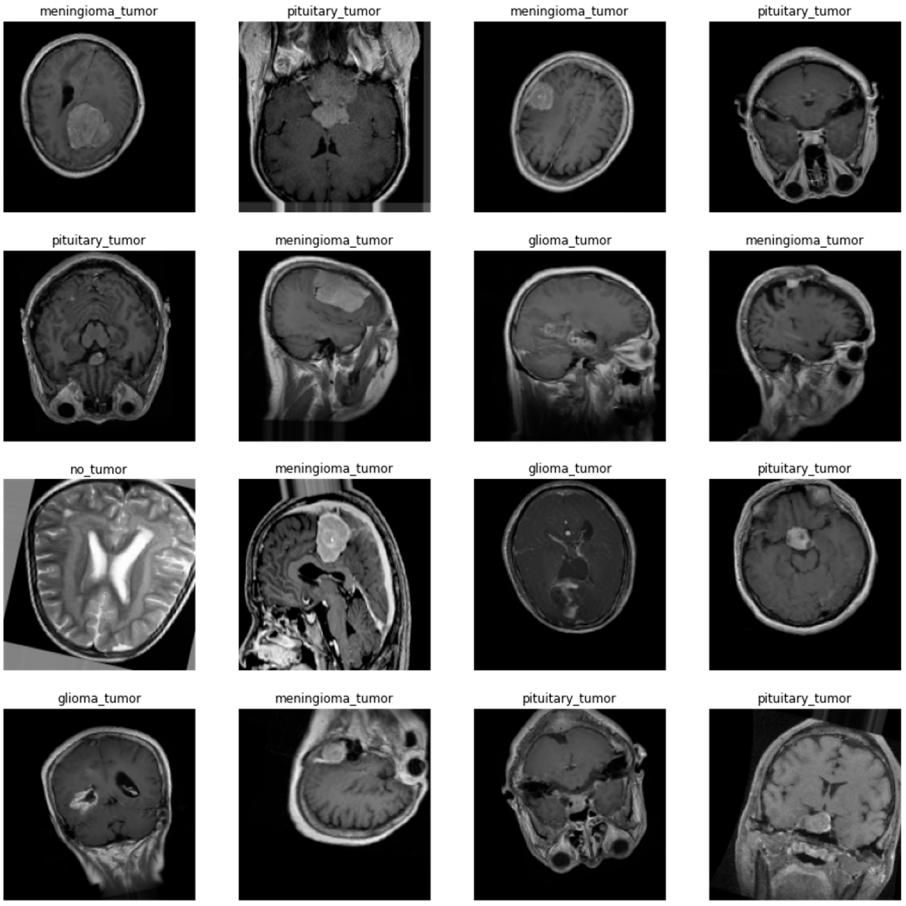

Figure 5. Samples from the training dataset with annotations. Augmented data.

2.4. Model Preparation and Training

Since the goal of the work was not only to achieve the highest accuracy rate possible in the task of brain cancer recognition but also to compare features of the algorithms described above, including ease of training, interpretability, and complexity, a few different versions of each model were trained. Two types of models were built for each type of network-Convolutional Neural Network and Vision Transformer. One created from scratch, trained only on the data described in Section 2.3, aiming to show how these algorithms handle training on small datasets and one pre-trained model fine-tuned on the target data, which should reveal which architecture has superior capabilities.

For each type of model, a few different architectures were tested, and only the results of the best were taken into consideration. The simple CNN was a network built of four convolutional layers intertwined with MaxPoolings and Dropouts, ending with a two-layer, flat classifier consisting of 128 and 4 neurons. In turn, in the case of a network trained with Transfer Learning methods, the 36-layer-deep Xception network was used as a base [33], ending with a Dropout and a two-layer classifier (512 and 4 neurons).

Both Vision Transformer models trained during the research were practically identical to the architecture shown in Figure 4; they differed mainly by size. The simple ViT had a latent space vector of 64 elements and was built of 3 encoder blocks (there were problems with training deeper networks) ending with a similar classifier as in the CNN model-128-neuron flat layer and final 4-neuron layer. For the Transfer Learning method, the model created by the architecture authors was used [16]-it consists of 32 encoder blocks and has a latent space of 1024 elements. The bigger ViT also ends with a two-layer classifier (512 and 4 neurons). Input images for both these models were divided into patches of 8×8 pixels.

The training process for all models was meant to be similar to make it easier to compare the results. The training went for 120 epochs using the RMSProp optimizer, Categorical

Crossentropy loss function, and Accuracy as a metric. To enhance the learning process, Label Smoothing Regularization was used [34], which proved to be essential while training the shallow ViT model. Graphical representations of architecture designs of CNN models, along with charts of training progress are shown in Figures A1–A4.

3. Results

3.1. Classification Results

While researching, various types of networks were trained, including different pre-trained architectures, with varying results. Some models were hard or even impossible to train on the dataset, while others achieved excellent results in a few first attempts. For every network type described in Section 2.4, the best prototype in terms of accuracy was chosen, and the results were gathered in Table 1. As the goal of the work was to compare the raw capabilities of the discussed architectures, no other parameters were taken into account while choosing top models. Although, along with the score achieved by the architecture, there is also a count of parameters the network had, as the algorithm's complexity is an essential factor while productizing the solution.

Table 1. Accuracies and parameters count for the best models.

Network Type	Accuracy [%]	Number of Parameters
Simple Conv Net	88.55	2,101,412
Pre-trained Xception	90.38	21,912,620
Simple Vision Transformer	90.18	3,466,564
Pre-trained Vision Transformer	92.30	303,734,276

The outcomes show that all tested architectures were able to provide classification accuracy close to 90%, which is comparable to the state-of-the-art solutions [11,13]. The pre-trained ViT model proved to be the best out of the tested architectures, but with a score only slightly better than a simple ViT or a pre-trained Xception model, at the cost of tremendous complexity, as shown in Table 1. This high complexity of the architecture can be explained by its generality; a Transformer is less suited to processing images than a CNN, however it can benefit more from larger datasets due to better data generalization. There are also smaller versions of the ViT available; the one considered here has 32 encoder blocks, but there is also one with 16 blocks, although, during this research, it turned out to be less accurate than its bigger version. Additional insights into classification results, along with confusion matrices for the models, can be found in Appendix A and Figures A5 and A6.

Noteworthy is the fact that the simple ViT demonstrated accuracy comparable with the Xception-based model, while not using many more parameters than a four-layer convolutional network. This property might make it a great computer vision algorithm for embedded devices, combining efficiency with relatively low demand for computing resources.

Regarding ease of training, the simple convolution model was the most straightforward to train. Reasonable results could have been obtained using almost any training parameters or layers layout, provided the network was kept relatively simple with some generalization mechanism (dropout, pooling, label smoothing). The Xception network was not much worse, although it required careful learning rate selection and many more resources. Only some types of CNNs worked so well during the research; models based on different pre-trained architectures were tried too, and some of them resulted in inferior outcomes or did not even manage to train. A few examples of such networks might be a VGG family or a ResNet family, which delivered results far from the top models.

Vision Transformers were more challenging to train than convolutional networks; to obtain a model with fair outcomes, numerous experiments needed to run. ViTs required more extensive generalization methods, slower learning, and bigger batches of images, increasing the training process's memory complexity. In addition, the size of the input patches significantly influences the network's performance, and it is hard to find the right heuristic to tell which size will yield the best results. Smaller patches allow the model to

extract more detailed local features, while discouraging attention to global patterns and notably increasing computational complexity.

3.2. Marking Points of Interest

In the case of this research, explaining the decisions of neural networks is even more important than the raw scores the models achieved. It not only provides insights that allow us to assess whether the network truly learned how to solve the task but also emphasizes areas of the image important to the diagnosis. Figures 6 and 7 present a few visualizations of classifications for simple versions of both types of architectures. The Grad-CAM for the CNN and Attention Rollout for the ViT were used to create the visualizations.

As can be seen in Figures 6 and 7, both types of networks produced reasonable heatmaps, although in some cases, it is hard to say what made the network predict a specific class. Both architectures had problems marking up reasons for a pituitary tumor prediction, but in other cases, they were able to tag some interesting features, especially in the case of a meningioma tumor, with neoplastic changes clearly outlined from the background. Additionally, both methods do not mark anything specific in the case of no-tumor pictures, thus avoiding unnecessary noise that might distract the doctor examining scans.

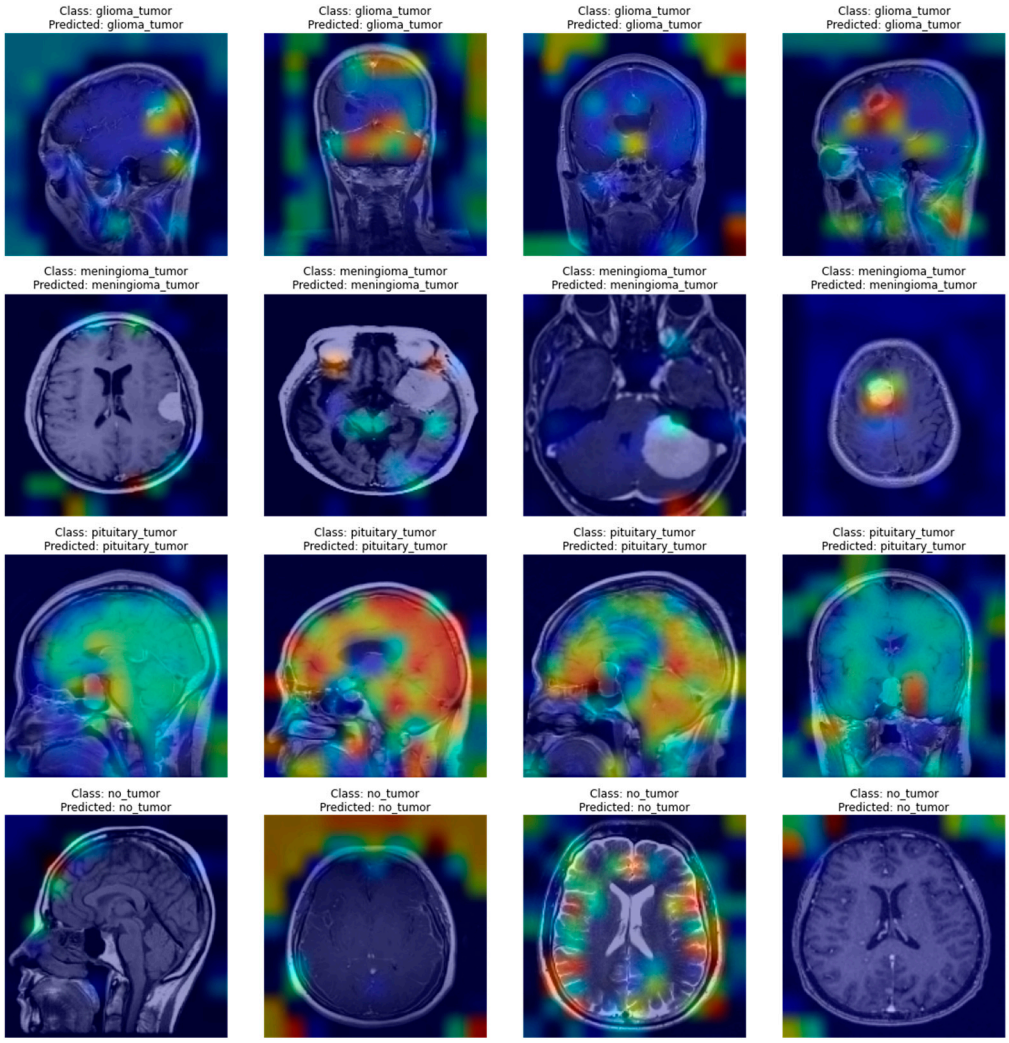

Figure 6. A few examples of Grad-CAM produced heatmaps for the simple CNN's decisions.

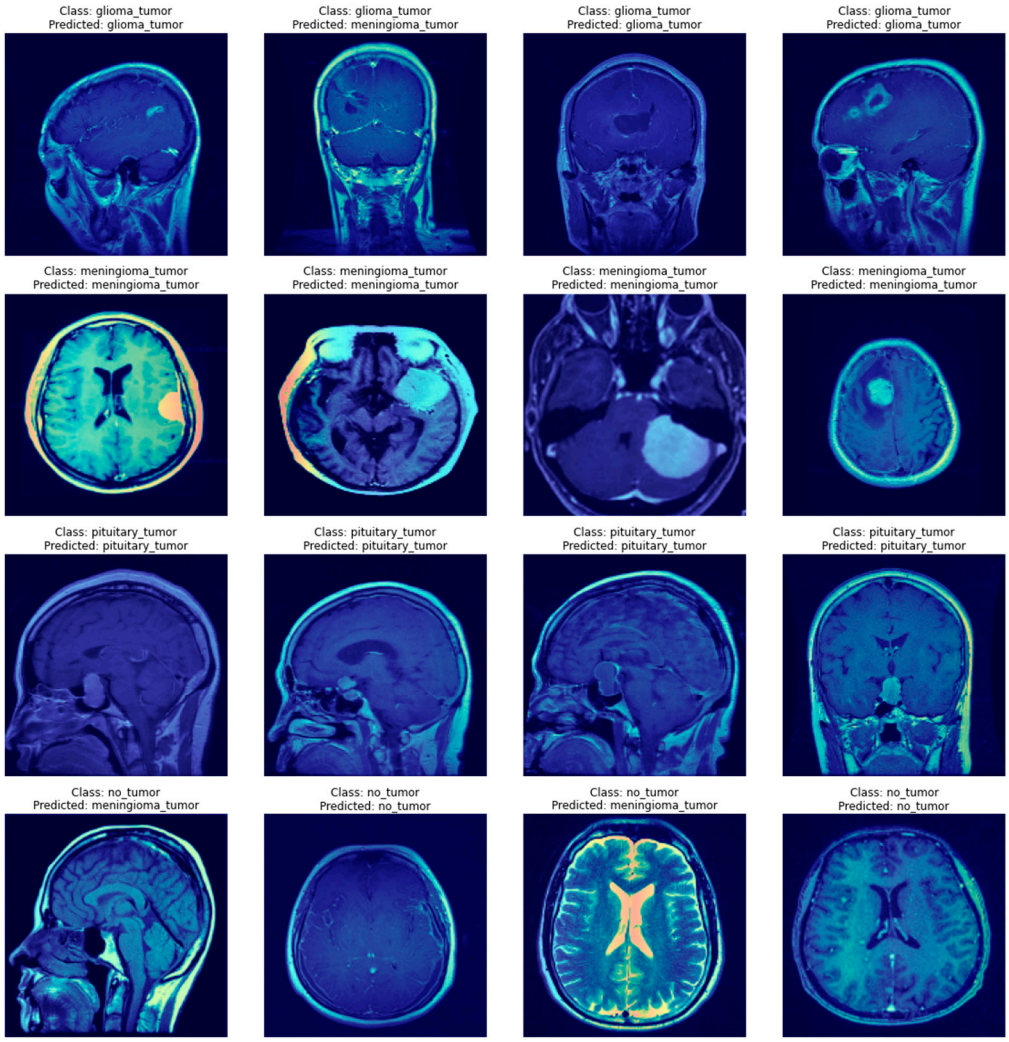

Figure 7. A few examples of Attention Rollout produces heatmaps for the simple ViT's decisions.

The results of both algorithms are of comparable quality as they both mark similar features of the image. Still, one has to notice that the outputs of ViT's visualizations are of better resolution, which is caused by the way the Attention Rollout casts the matrices weights by all the linearly scaling matrices of encoders, thus producing the outputs shaped similar to the network's input. In contrast, the Grad-CAM produces a heatmap of the resolution of a chosen feature map's dimensionality, which might differ between networks, and, in some cases, not provide satisfying outcomes. Such an example might be the Xception model, where the last convolutional layer produces feature maps of the 5×5 size, which does not provide any valuable insights. Furthermore, the ViT model's visualizations show that it might be better in ignoring the background, which causes the output images to look less noisy.

It is noteworthy that these visualizations are only one of the many possible. There are different visualization algorithms, and they can be combined to achieve superior results. In addition, the output images shown in Figures 6 and 7 are class-specific, which means that for every one of these images, a few more visualizations might be created, which would allow for checking whether anything was omitted while making a diagnosis.

4. Discussion

Both Convolutional Neural Networks and Vision Transformers were able to solve the task of brain cancer classification based on the MRI scans and achieved an accuracy of approximately 90%, which is close to the state-of-the-art methods. In combination with reasonable looking heatmaps indicating the reason for decisions, this performance allows the inference that building CAD systems helping with brain tumor diagnosis based on such artificial intelligence algorithms is possible.

Attention-based models proved well suited for the task and even outperformed traditional CNNs at the cost of complexity. However, even the model comparably simple to the four-layer CNN handled the problem to a similar degree as one of the most complex CNN models to date. ViTs proved to be harder to train than CNNs, requiring a greater amount of data and additional precautions during the training process, which makes the accuracy gain costly, despite a shorter training time. As a bonus to the excellent scores achieved by ViTs, they also provide exceptionally detailed heatmaps that can be used for diagnosis support. All of these factors show that Vision Transformer is the architecture that should be further developed in the context of computer vision as it is a powerful and general model whose possibilities have yet to be fully explored.

To enhance opportunities for ViT-based CAD systems, there is a need for more extensive and high-quality datasets of labeled MRI images, which might allow experimenting with different types of architectures. Models using attention mechanisms can be further upgraded with the use of hybrid methods, different attention calculations, novel initialization and optimization methods, or adaptive attention. In addition, further research could result in a wider variety of available pre-trained ViT models, which could help to productize the solution to different IoT fields, including medicine.

Summarizing the conclusions gathered here, computer-aided diagnosis systems based on neural networks producing classification labels and heatmaps are possible to create. They might prove helpful in speeding up and raising the quality of a diagnosis and lowering the risk of overlooking neoplastic changes. Vision Transformer models and other attention-based architectures seem to be excellent choices for this task. Presented techniques can be also applied for semantic pattern classification in other IoT areas [35].

Author Contributions: J.K.: conceptualization, methodology, software, validation, formal analysis, investigation, writing—original draft, visualization. M.R.O.: conceptualization, methodology, validation, writing—review and editing, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the AGH University of Science and Technology Research, Grant No 16.16.120.773.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used for training can be acquired from [9]. Other data available on request from authors.

Conflicts of Interest: The authors have no conflict of interest to declare for this manuscript.

Appendix A

Figures A1 and A2 show the exact architectures of CNN models described in the work. As can be seen, the convolutional model from Figure A1 is a relatively simple neural network featuring only four convolutional layers and an extensive set of normalization and generalization mechanisms. On the other hand, the Xception model, which is visualized in Figure A2, is a much more extensive network, making use of residual connections and parallel convolutions. Such a design allows the architecture to extract features from the data on multiple levels, not only from the close proximity of a pixel.

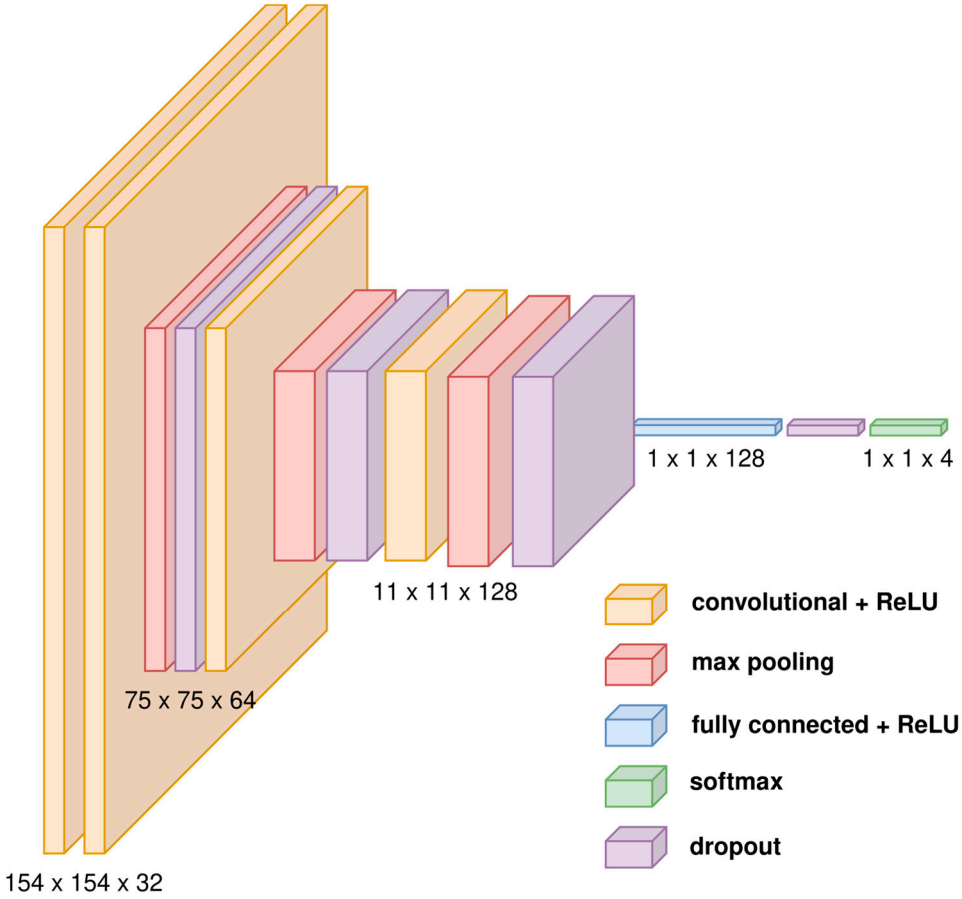

Figure A1. Visualization of the simple convolution model's architecture.

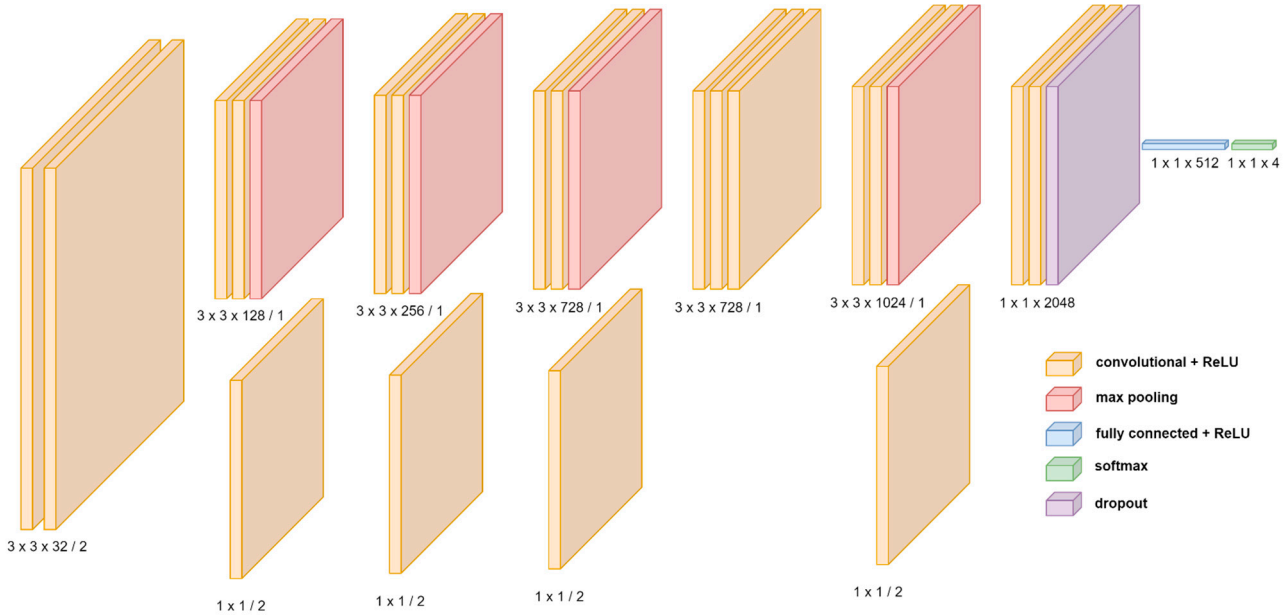

Figure A2. Visualization of the Xception model architecture.

Figures A3 and A4 present the training progress for all four models discussed in the work. As can be seen, the simple Vision Transformer training chart is much more jagged in comparison with other architectures, which results from the low stability of its learning process and high demand for data. It is worth noticing that no such thing happens in the case of the pre-trained ViT as it was already trained with enormous quantities of training samples. At the same time, the charts of the pre-trained ViT model do not flatten so much as both convolutional networks do, leaving space for even more fine-tuning.

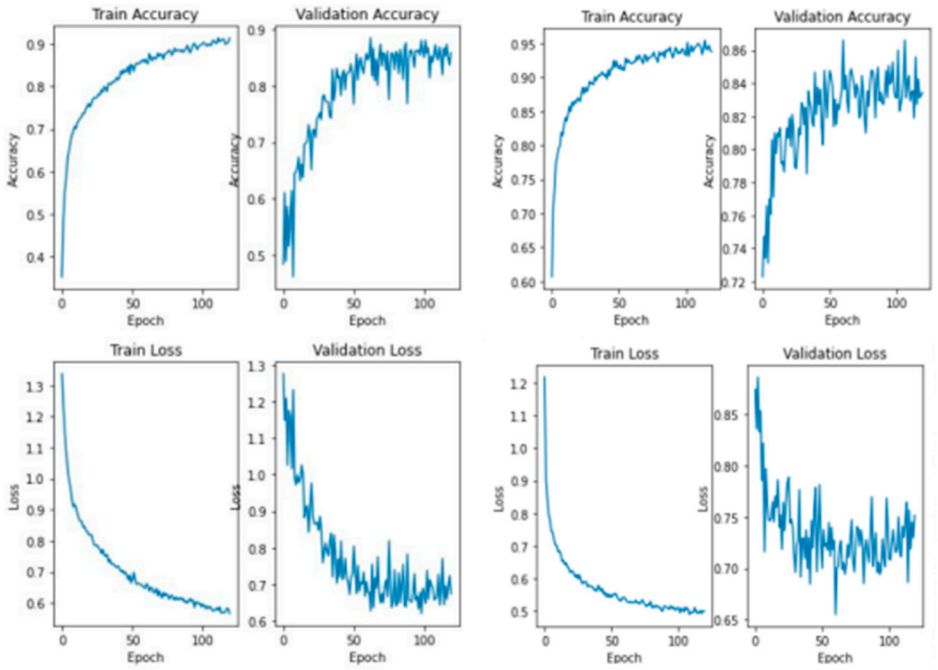

Figure A3. Training progress of the convolution-based models–simple Convolution (**left**) and Xception (**right**).

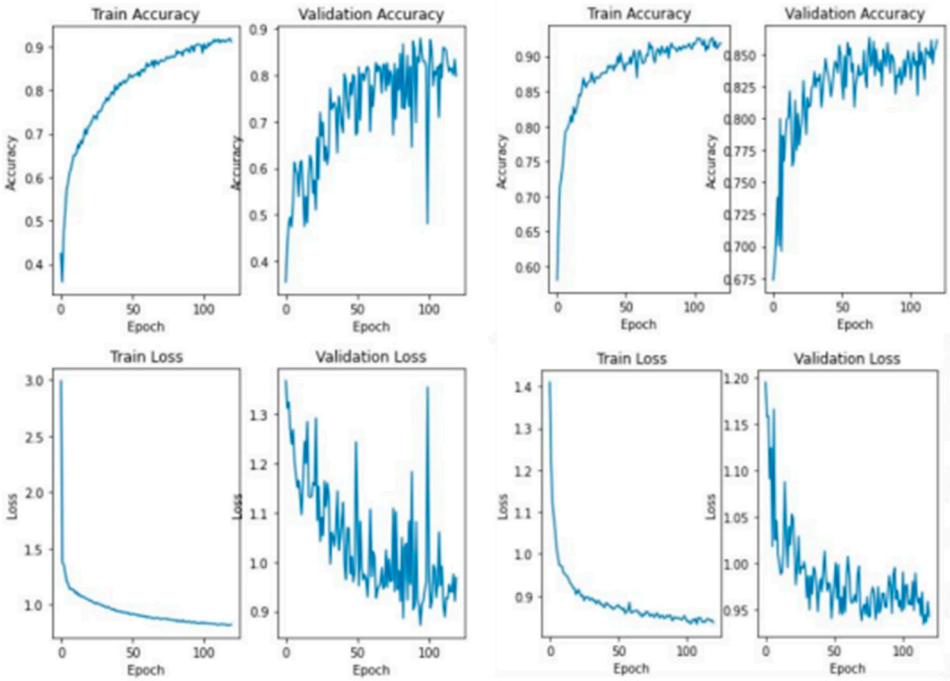

Figure A4. Training progress of the attention-based models: simple Vision Transformer (**left**) and pre-trained Vision Transformer (**right**).

Confusion Matrices presented on Figures A5 and A6 (symbols on axes represent image classes: G–Glioma, M–Meningioma, P–Pituitary, N–no tumor) show that none of the architectures proved to behave differently regarding different neoplastic lesion types. Error distribution is very similar for every trained model, with meningiomas being relatively frequently mistaken with gliomas (and conversely) and pituitary tumors. At the same time, the rate of false positive predictions is low for all the networks.

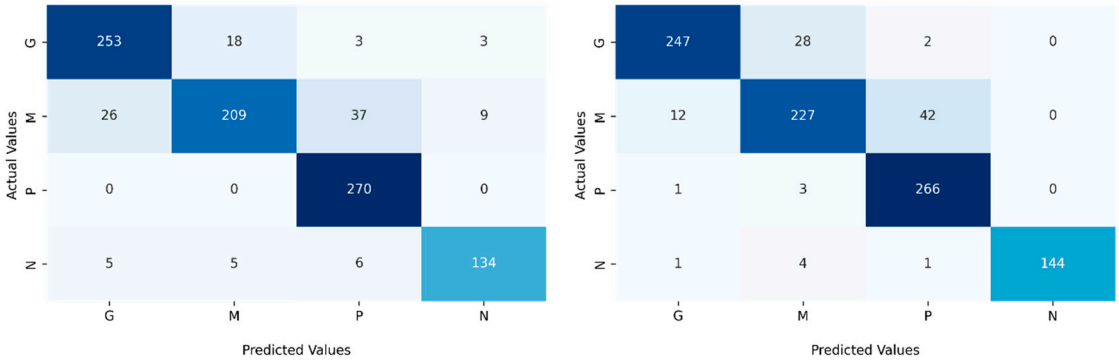

Figure A5. Confusion Matrices for simple Convolutional Network (**left**) and Xception-based network (**right**).

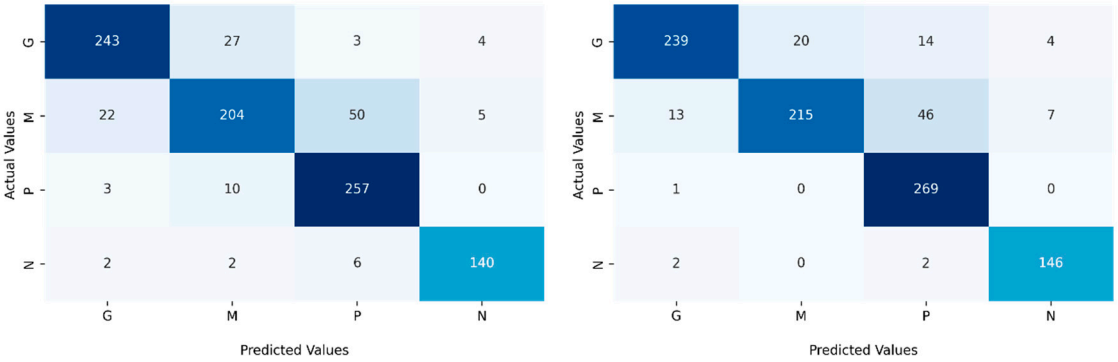

Figure A6. Confusion Matrices for simple Vision Transformer (**left**) pre-trained Vision Transformer (**right**).

References

1. OECD. *Health at a Glance 2021: OECD Indicators*; OECD Publishing: Paris, France, 2021. [CrossRef]
2. Ferlay, J.; Colombet, M.; Soerjomataram, I.; Parkin, D.M.; Piñeros, M.; Znaor, A.; Bray, F. Cancer Statistics for the Year 2020: An Overview. *Int. J. Cancer* **2021**, *149*, 778–789. [CrossRef] [PubMed]
3. Brain Tumor. Available online: https://www.cancer.net/cancer-types/brain-tumor (accessed on 6 October 2022).
4. Ostrom, Q.T.; Cioffi, G.; Waite, K.; Kruchko, C.; Barnholtz-Sloan, J.S. CBTRUS Statistical Report: Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2014–2018. *Neuro-Oncol.* **2021**, *23*, iii1–iii105. [CrossRef]
5. Miller, K.D.; Ostrom, Q.T.; Kruchko, C.; Patil, N.; Tihan, T.; Cioffi, G.; Fuchs, H.E.; Waite, K.A.; Jemal, A.; Siegel, R.L.; et al. Brain and Other Central Nervous System Tumor Statistics, 2021. CA. *Cancer J. Clin.* **2021**, *71*, 381–406. [CrossRef]
6. Louis, D.N.; Perry, A.; Wesseling, P.; Brat, D.J.; Cree, I.A.; Figarella-Branger, D.; Hawkins, C.; Ng, H.K.; Pfister, S.M.; Reifenberger, G.; et al. The 2021 WHO Classification of Tumors of the Central Nervous System: A Summary. *Neuro-Oncol.* **2021**, *23*, 1231–1251. [CrossRef] [PubMed]
7. Floca, R. Challenges of Open Data in Medical Research. In *Opening Science*; Bartling, S., Friesike, S., Eds.; Springer: Cham, Switzerland, 2014. [CrossRef]
8. Pampel, H.; Dallmeier-Tiessen, S. Open Research Data: From Vision to Practice. In *Opening Science: The Evolving Guide on How the Internet is Changing Research, Collaboration and Scholarly Publishing*; Bartling, S., Friesike, S., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 213–224. ISBN 978-3-319-00026-8.
9. Bhuvaji, S.; Kadam, A.; Bhumkar, P.; Dedge, S.; Kanchan, S. Brain Tumor Classification (MRI). *Kaggle* **2020**, *10*. [CrossRef]
10. Filler, A. The History, Development and Impact of Computed Imaging in Neurological Diagnosis and Neurosurgery: CT, MRI, and DTI. *Nat. Preced.* **2009**, 3–7, 16–30, 51–55. [CrossRef]

11. Cheng, J.-Z.; Ni, D.; Chou, Y.-H.; Qin, J.; Tiu, C.-M.; Chang, Y.-C.; Huang, C.-S.; Shen, D.; Chen, C.-M. Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans. *Sci. Rep.* **2016**, *6*, 24454. [CrossRef] [PubMed]
12. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; ISBN 978-0-262-03561-3.
13. Chatterjee, S.; Nizamani, F.A.; Nürnberger, A.; Speck, O. Classification of Brain Tumours in MR Images Using Deep Spatiospatial Models. *Sci. Rep.* **2022**, *12*, 1505. [CrossRef]
14. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]
15. Gao, J.; Jiang, Q.; Zhou, B.; Chen, D. Convolutional Neural Networks for Computer-Aided Detection or Diagnosis in Medical Image Analysis: An Overview. *Math. Biosci. Eng.* **2019**, *16*, 6536–6561. [CrossRef]
16. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**. [CrossRef]
17. Akkus, Z.; Galimzianova, A.; Hoogi, A.; Rubin, D.L.; Erickson, B.J. Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. *J. Digit. Imaging* **2017**, *30*, 449–459. [CrossRef] [PubMed]
18. Biscione, V.; Bowers, J.S. Convolutional Neural Networks Are Not Invariant to Translation, but They Can Learn to Be. *arXiv* **2021**. [CrossRef]
19. Qin, Z.; Yu, F.; Liu, C.; Chen, X. How Convolutional Neural Network See the World—A Survey of Convolutional Neural Network Visualization Methods. *arXiv* **2018**. [CrossRef]
20. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**. [CrossRef]
21. Cheng, J.; Dong, L.; Lapata, M. Long Short-Term Memory-Networks for Machine Reading. *arXiv* **2016**. [CrossRef]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**. [CrossRef]
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**. [CrossRef]
24. Brauwers, G.; Frasincar, F. A General Survey on Attention Mechanisms in Deep Learning. *IEEE Trans. Knowl. Data Eng.* **2021**. [CrossRef]
25. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**. [CrossRef]
26. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional Sequence to Sequence Learning. *arXiv* **2017**. [CrossRef]
27. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention Mechanisms in Computer Vision: A Survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
28. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *arXiv* **2014**. [CrossRef]
29. Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation Networks for Object Detection. *arXiv* **2017**. [CrossRef]
30. Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating Long Sequences with Sparse Transformers. *arXiv* **2019**. [CrossRef]
31. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**. [CrossRef]
32. Abnar, S.; Zuidema, W. Quantifying Attention Flow in Transformers. *arXiv* **2020**. [CrossRef]
33. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2016**. [CrossRef]
34. Xu, Y.; Xu, Y.; Qian, Q.; Li, H.; Jin, R. Towards Understanding Label Smoothing. *arXiv* **2020**. [CrossRef]
35. Ogiela, L.; Tadeusiewicz, R.; Ogiela, M.R. Cognitive Analysis In Diagnostic DSS-Type IT Systems. *Lect. Notes Artif. Intell.* **2006**, *4029*, 962–971. [CrossRef]