Medical Image Diagnosis Using Convolutional Neural Network: Transfer Learning and Visualization of the Decision Making

Juuso Korhonen

School of Electrical Engineering

Bachelor's thesis
Espoo 09.05.2021

Supervisor

Docent Samuli Aalto

Advisor

Associate Professor Esa Ollila

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the bachelor's thesis

Author Juuso Korhonen

Title Medical Image Diagnosis Using Convolutional Neural Network: Transfer Learning and Visualization of the Decision Making

Degree programme Electronics and electrical engineering

Major Information technology **Code of major** ELEC0007

Teacher in charge Docent Samuli Aalto

Advisor Associate Professor Esa Ollila

Date 09.05.2021 **Number of pages** 34+1 **Language** English

Abstract

The purpose of this thesis is to study the suitability of convolutional neural networks (CNN) for medical image analysis focusing on diagnosis of magnetic resonance imaging (MRI) brain scan. The thesis identifies and provides solutions for the two key problems encountered when introducing CNNs into the field: data scarcity and explainability of the decision-making. For data scarcity, the thesis investigated the use of transfer learning. To increase the explainability, the gradient-based class activation mapping (Grad-CAM) was used. These methods were tested on a classification task with a brain tumor dataset. Transfer learning was found to make the training converge faster and to increase the classification accuracy with unseen test data compared to random initialization of the parameters of the model (88% accuracy vs. 76% accuracy). With correctly classified tumor images, Grad-CAM was found to give indication of the region the model based its classification on. The visualizations also provided some indication of the generalizability of the model's classification, as the visualizations were slightly better with the transfer learning model. However, the resolution and the boundary information were found to be poor in the visualizations, and with falsely classified images, the method offered no reasonable explanation for why the classification failed. Transfer learning was found to be a viable solution to alleviate the problems of data scarcity. The Grad-CAM method was found to be a good first step in explaining the CNN decision-making, but further development is needed to achieve a finer detail level.

Keywords MRI classification, CNN, Grad-CAM, CNN visualization, Transfer Learning

Aalto University
School of Electrical
Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Tekniikan kandidaatintyön tiivistelmä

Tekijä Juuso Korhonen

Työn nimi Medical Image Diagnosis Using Convolutional Neural Network: Transfer Learning and Visualization of the Decision Making

Koulutusohjelma Electronics and electrical engineering

Pääaine Information technology **Pääaineen koodi** ELEC0007

Vastuuopettaja ja ohjaaja Docent Samuli Aalto

Päivämäärä 09.05.2021 **Sivumäärä** 34+1 **Kieli** Englanti

Tiivistelmä

Neuroverkot ja syväoppiminen ovat herättäneet suurta kiinnostusta lukuisilla eri tieteen ja teollisuuden aloilla viime vuosina. Samaan aikaan lääketieteellisen kuvantamisen kysynnän kasvu ja radiologien ylityöllistyminen ovat herättäneet tarpeen automatisoida lääketieteellisten kuvien analyysia. Eri tehtäviin soveltuvista ja erilaisista neuroverkkoarkkitehtuureista, konvoluutioverkot ovat osoittautuneet erityisen soveltuviksi kuva-analyysiin.

Tämän kandityön tarkoituksena on tutkia konvoluutioverkkojen soveltuvuutta lääketieteelliseen kuva-analyysiin keskittyen magneettikuvattujen aivokuvien diagnosointiin. Kandityö pyrkii esittelemään ratkaisuja ongelmiin, joihin törmätään, kun konvoluutioverkkoja yritetään soveltaa tällä kentällä. Kandityö keskittyy kahteen ongelmaan: datan hankkimisen vaikeuteen ja konvoluutioverkkojen päätöksenteon selitettävyyteen.

Datan hankkiminen lääketieteellisen kuvantamisen kentällä on edelleen haastavaa ja paljon resursseja vaativaa. Tämä muodostaa haasteen konvoluutioverkkojen soveltuvuudelle, sillä ne tarvitsevat toimiakseen tyypillisesti paljon dataa opetusprosessia varten.

Päätöksenteon selitettävyydellä tarkoitetaan sitä, että konvoluutioverkon aivokuvan perusteella antama diagnoosi tulisi pystyä jollain tapaa jäljittää takaisin aivokuvaan ja sen eri osiin ihmiselle ymmärrettävässä muodossa. Tämä lisää diagnoosin luotettavuutta ja toimii usein edellytyksenä näiden työkalujen soveltuvuudelle kliiniseen käyttöön.

Datan hankkimisen vaikeudesta koituvien ongelmien lieventämiseksi kandityössä tutkittiin transfer learning nimistä menetelmää. Ideana menetelmässä on hyödyntää jo opittua asiaa uuden asian oppimisessa. Kuvien tapauksessa tämä tarkoittaa sitä, että ne rakentuvat universaaleista muodoista kuten reunoista, oli kyseessä sitten luonnollinen kuva, kuten koiraa esittävä kuva, tai aivokuva. Tämä mahdollistaa luonnollisilla kuvilla esiopetetun konvoluutioverkon hyödyntämisen aivokuvien diagnosoinnissa.

Tehdäkseen päätöksenteosta selitettävämpää kandityö tutki gradienttipohjaista visualisointimenetelmää nimeltään gradienttipohjainen luokka-aktivointi-menetelmä (engl. Gradient-based class activation mapping, Grad-CAM). Menetelmässä on ideana ulottaa gradienttitarkastelu verkon sisäisiin esityksiin syötetystä kuvasta. Tarkoituksena menetelmässä on saada selville konvoluutioverkon luokittelupäätökseen eniten vaikuttavat alueet kuvasta.

Näitä metodeja testattiin kandityössä magneettikuvantamisella hankitulla aivosyöpädatasetillä. Transfer learning -menetelmä onnistui lyhentämään opettamiseen vaadittavaa aikaa ja parantamaan luokittelutarkkuutta verkolle ennennäkemättömän testidatan kanssa. Luokittelutarkkuus parani testidatassa 76%:sta 88%:iin, kun luonnollisilla kuvilla esiopetettua mallia verrattiin samaan malliin, jossa parametrit olivat satunnaisalustettuja.

Oikein luokiteltujen kuvien kanssa Grad-CAM-metodi onnistui antamaan viitteitä alueesta, jolle verkko perusti luokittelupäätöksensä (ts., jolla verkko arveli syöpäkudoksen sijaitsevan). Visualisoinnit antoivat myös hiukan viitteitä mallin luokittelun yleistettävyydestä, sillä visualisoinnit olivat hiukan parempia esiopetetun mallin kanssa verrattuna satunnaisalustetun mallin kanssa saatuihin visualisointeihin. Kuitenkin resoluutio ja rajainformaatio (erottelu terveen ja syöpäkudoksen välillä näiden raja-alueella) olivat visualisoinneissa yleisesti huonoja. Virheluokittelujen kohdalla visualisointimenetelmä ei myöskään auttanut ymmärtämään miksi luokittelu meni väärin.

Transfer learning osoittautui varteenotettavaksi menetelmäksi vähentää opetusvaatimuksia konvoluutioverkkojen kohdalla (vähemmän aikaa ja parempi luokittelutarkkuus kuin verkon parametrien satunnaisella alustamisella). Grad-CAM metodi osoittautui myös hyväksi ensiaskeleeksi verkon päätöksenteon ymmärryksessä, mutta lisää kehitystä tarvitaan metodin parissa, jotta visualisoinneista saadaan yksityiskohtaisempia. Kandityössä arvioitiin, että visualisointien huono resoluutio ja rajainformaatio johtuvat verkon sisäisten esitysten matalasta resoluutiosta ja sijainti-informaation menetyksestä verkon saapuessa niihin.

Avainsanat MRI classification, CNN, Grad-CAM, CNN visualization, Transfer Learning

Contents

1 Introduction

In recent years the use of neural networks and deep learning has seen a lot of interest in different fields of science and industry. There exist different kinds of architectures of neural networks, which have managed to outperform other machine learning methods on various tasks ranging from object recognition and classification to natural language processing. One of these architectures is called convolutional neural network (CNN), and it has been found especially suitable for analyzing images.

Medical image analysis is one of the most compelling applications for CNNs. Tasks like image classification and image segmentation are laborious and can take up a lot of working hours from medical experts. The average time for a radiologist to complete a diagnosis of a magnetic resonance imaging (MRI) head scan is about 18 minutes as shown in figure A1. An increase in the number of examinations can result in either less time for the radiologist to do the analysis or longer working shifts, which could, in turn, make the analysis more error-prone. Using CNNs to guide and automate this process can make it much more efficient and increase the quality of the work.

CNN-based methods have provided state-of-the-art results in tasks such as melanoma recognition [1] and Alzheimer's disease detection [2]. However, the use of these methods in clinical work is still in its infancy, since they suffer from the same lack of explainability and thus lack of reliability as other machine learning methods. In the case of neural networks, this is even more of an issue, because of the lack of theoretical understanding of how these methods work. Also, the difficulty of data acquisition in the medical field has for a long time been a problem in using these methods, which require a lot of data.

The purpose of this thesis is to study the suitability of CNNs for medical image analysis focusing on MRI brain scan diagnosis. The thesis aims to provide solutions for the two key problems encountered when introducing CNNs into the field: data scarcity and explainability of the decision-making.

Acquiring data in the field of medical imaging is difficult and resource-intensive. This poses a challenge to the suitability of CNNs, as they typically need a lot of data for the training process. Explainability of the decision-making means that a diagnosis made on the basis of a brain image should be able to be traced back to the brain image in a human-understandable form. This increases the reliability of the diagnosis and is often a prerequisite for the suitability of these tools for clinical use.

To alleviate the problems arising from the difficulty of obtaining data, a method called transfer learning is studied in the thesis. The idea of the method is to utilize what has already been learned in learning a new thing. In the case of images, this means that they are constructed from universal shapes such as edges, be it a natural image, such as an image of a dog, or a brain image. This allows the use of a CNN pre-trained with natural images in the diagnosis of brain images.

To make the decision-making of the CNN more explainable, the thesis examines a gradient-based visualization method called Gradient-based class activation mapping (Grad-CAM). The idea of the method is to extend the gradient analysis to the network's internal representations of the input image. The purpose of the method is to find out the areas of the image that most influence the classification decision of the CNN. The transfer learning and visualization methods are tested with a brain tumor data set obtained by magnetic resonance imaging, which is one of the most common modalities of medical imaging.

The structure of this thesis is the following: background section will cover the necessary theory of deep learning to understand CNNs and their suitability for medical image analysis. It will also discuss methods to visualize CNN decision-making and the method of transfer learning. In the data and methods section, the brain tumor dataset is presented and the classification and visualization pipeline is specified. In the results section, transfer learning, classification, and visualization results are presented and discussed. In the conclusions section the results will be compared against the research questions: do the proposed methods (Grad-CAM and transfer learning) help with the problems of explainability and data scarcity when introducing CNNs into medical image analysis?

2 Background

This section will first briefly go through the recent developments and challenges in using machine learning in medical image analysis. The high dimensionality of the medical images is found to be the main challenge in using traditional machine learning methods, like support vector machines (SVM), in the medical image analysis. Convolutional neural networks (CNN) from the deep learning methods are offered as a solution since CNNs have proven to be able to solve complex high-dimensional problems. Chapter 2.2 will present the basics of deep learning, which are necessary to understand the basic structure of a CNN, which is described in chapter 2.3. Chapter 2.4 will present the method of transfer learning, and chapter 2.5 will discuss two main techniques to visualize the decision-making of the CNN: occlusion-based and gradient-based techniques. A gradient based visualization technique and a transfer learning model are used in the empirical section of this thesis to help with the introduction of CNNs into the medical image analysis.

2.1 Medical image analysis and machine learning

For the past few decades, medical image analysis has seen continuous growth in demand. For example, among older adults, computed tomography (CT) imaging rates were 428 per 1000 persons in 2016 vs 204 per 1000 persons in 2000 in US health care systems [3]. Although improvements in imaging techniques and computational hardware have led to a decrease in the actual scan duration across modalities, the analysis of the image, especially if done solely by a human radiologist, can quickly lead to a bottleneck in the process.

This has led to an increasing interest to automate this analysis process. Although various machine-learning models have been tried for automated medical image analysis, two of the most prominent approaches are support vector machine (SVM)-based and deep learning-based models [4][5].

Medical image analysis has still remained a complicated problem due to the complexity of the medical images. A successful classification approach should be able to recognize often subtle differences between the classes among a huge amount of possible features. Using the raw pixel intensities as features for a classification problem one quickly runs into the curse of dimensionality problem. This is why for most traditional machine learning algorithms, including SVMs, some kind of feature extraction method is required.

SVMs tend to rely on heavy pre-processing and hand-crafted features as they are not able to extract adaptive features from the raw data. With brain MRI images, the approach often includes data preprocessing methods for removing skull and neck tissue and segmentation into different types of brain tissue followed by feature extraction with principal-component-analysis (PCA), wavelet transform, or other transforms. Despite their popularity, SVMs have been criticized for the poor performance on the raw data and for the expert knowledge needed both in algorithm development and in the task domain to get working solutions. It is also hard to know in advance which features are important for correct classification.

In contrast, modern deep learning models allow the use of raw data as input by automatically extracting features, which are adaptive to the given training data set. This end-to-end learning is a key advantage in deep learning approaches and allows the optimization of all the steps of the task based on the training data. This has the potential to lead to optimal performance. However, deep learning approaches typically require lots of data to achieve good performance and with smaller datasets other machine learning methods like SVMs can produce better results.

2.2 Deep learning

With the rapid increment in the processing power and in the amount of collected data, we have seen the revolution of deep learning algorithms. As a subset of machine learning algorithms, deep learning algorithms have proven to be able to solve complex problems. The early success of these algorithms was witnessed in several popular image analysis benchmarks, perhaps the most famous one being when a deep learning model (a CNN) halved the second-best error rate on the ImageNet Large-Scale Visual Recognition Challenge in 2012 [6].

Deep learning algorithms have not stayed limited to image analysis but are producing state-of-the-art solutions in areas like natural language processing, gameplay, and time-series analysis [7]. There exist different kinds of deep learning architectures for different tasks, but they all rely on the principles that are set by the artificial neural network.

2.2.1 Artificial neural network

Artificial neural networks, usually called simply neural networks, are the main computational systems behind deep learning. The core computational component of neural networks, the artificial neuron, was first modeled in 1958 by Frank Rosenblatt [8], taking inspiration from the functioning of a real neuron found in the brain.

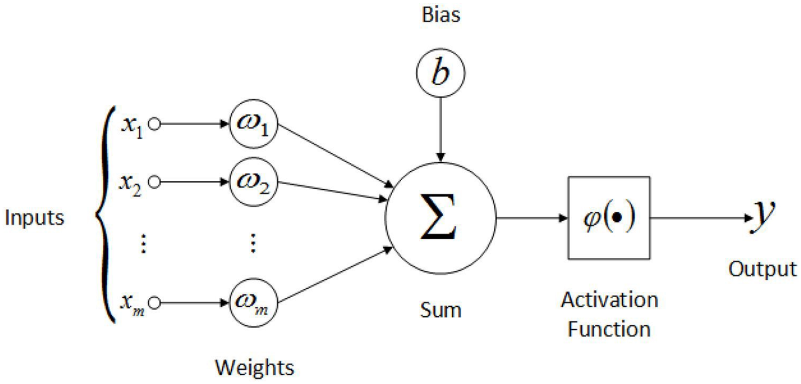


Figure 1: Structure of an artificial neuron.

$$y = \varphi\left(\sum_{j=0}^{m} w_j x_j + b\right), \qquad y, x_j, w_j, b \in \mathbb{R}. \tag{1}$$

The function of an artificial neuron, visualized in figure 1, processes input X to output Y. The input elements x_j can be outputs of other neurons or, for example, pixel intensity values of an image. The input elements x_j have corresponding weights w_j. The weights imitate the synaptic strengths of a real neuron. To compute the output y, all the input elements are multiplied with their corresponding weights and then summed together. Adding the bias b to this sum and feeding the result through activation function forms the output y, as described in the equation (1). The activation function φ and bias b affect what is needed to activate the neuron. The activation function φ is typically some kind of a nonlinear threshold function like sigmoid (shown in figure 2) or rectified linear unit (ReLU) (shown in figure 3), and the bias moves the threshold of this function. The nonlinearity of the activation function is a key element in being able to define nonlinear decision boundaries to the input space.

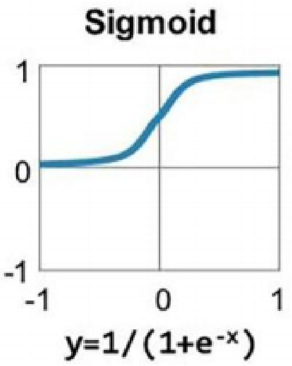


Figure 2: The plot and function of a sigmoid activation function. Adding the bias would move threshold limit which is at $x = 0$ initially.

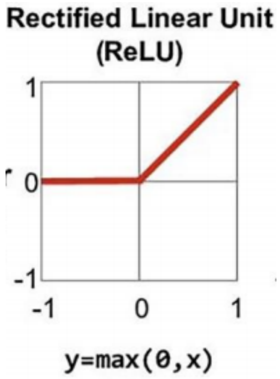


Figure 3: The plot and function of a ReLU activation function. Adding the bias would move threshold limit which is at $x = 0$ initially.

Multiple concatenated neurons form a layer of a neural network. An architecture consisting of an input layer, an output layer, and at least one intermediate layer, referred to as a hidden layer, is called a multi-layer perceptron (MLP). Figure 4

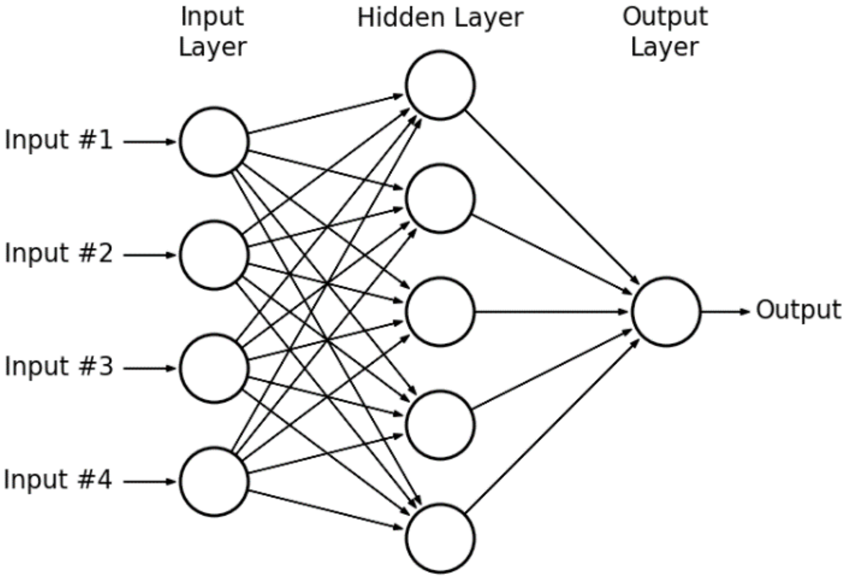


Figure 4: Structure of an artificial neural network.

shows this kind of architecture. A key feature of the architecture is that the output of every neuron from the previous layer is connected to every neuron in the following layer. The multilayered structure has the ability to learn increasingly more complex representations, approaching a universal function approximator with an arbitrary amount of layers [9]. The added layers, however, come with an added computational cost and overparametrization issues.

The outputs of the neurons in the final layer produce the output of the network. In classification tasks, instead of using ReLU or sigmoid, a softmax function is often used as an activation function φ for the final layer:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{2}$$

where K is the number of neurons in the output layer (i.e., the number of classes). Softmax scales the outputs so that they all add up to one and can be intuitively interpreted as probabilities for different classes.

2.2.2 Training the network

To train the network, we need labeled data. One way to label the data is the so-called "one-hot-encoding". In this method, the label for each class is a vector with the same length as the number of classes in the dataset. So in the example of classifying MRI head scans to either belonging to a person with the disease or a healthy person, the vector would have two digits. Each digit represents a class so that for example the first digit represents the presence of a disease in the image and the second represents the image being a healthy image. So the vectors would be [1, 0] for images of a person with the disease, and [0, 1] for images of a healthy person.

The neural network for this classification task would have two output neurons in the final layer that predict the likelihood of the input image belonging to each of the classes. Processing the input of a neural network is called forward propagation. The resulting vector of the forward propagation of an image of a person with the disease (target: $Y = [1,\ 0]$) could look like this after the softmax activation in the last layer: $\hat{Y} = [0.8, 0.2]$;

While training, we try to make the output of the network to be as similar as possible to the target output. This happens through minimizing the loss of the network which we calculate with a suitable loss function. In classification tasks, the used loss function is usually cross-entropy loss defined by:

$$C(Y, \hat{Y}) = -\sum_{x \in X} Y(x) log \hat{Y}(x). \tag{3}$$

The cross-entropy loss function measures the dissimilarity between the output probability distribution $\hat{Y}$ and the target output probability distribution Y. Note that for C([1, 0], [1, 0]), i.e. for correct classification, the loss would be zero and minimized. In our example, the cross-entropy loss is calculated in the following way:

$$C([1,0],[0.8,0.2]) = -1 * \log(0.8) - 0 * \log(0.2) = 0.2231. \tag{4}$$

The loss values are passed to the optimizer of the neural network, which uses backpropagation to determine the amount in which the weights of the network have to be tuned. A modern overview of the backpropagation is given in [10]. In backpropagation, the gradient of the loss function is calculated by deriving the partial derivatives of the loss with respect to the tunable parameters of the network. Instead of doing it the naive way by calculating derivatives one by one, it is done efficiently layer by layer by using transpose matrix multiplications and the chain-rule:

$$\delta^l := (f^l)' \cdot (W^{l+1})^T \cdot \dots \cdot (W^{L-1})^T \cdot (f^{L-1})' \cdot (W^L)^T \cdot (f^L)' \cdot \nabla_{a^L} C. \tag{5}$$

where the δ^l is interpreted as the error at level l, L is the number of layers in the network, W^l is the weight matrix at level l and the $(f^l)'$ is the vector containing derivatives of the activation functions at layer l. The gradient of the weights at layer l is then:

$$\nabla_{W^l} C = \delta^l (a^{l-1})^T \tag{6}$$

where a^{l-1} is the vector containing the activations (i.e., the outputs of the neurons) in the previous layer. How the gradients are used to update the parameters depends on the chosen optimizer. Stochastic gradient descent (SGD), for example, calculates an estimate of the gradient of the loss function by averaging the gradients calculated for a randomly selected subset X of the data rather than the whole dataset. This reduces the computational cost of updating the parameters. The actual update happens by multiplying this gradient estimate with the learning rate α, and subtracting it from the corresponding parameters, as described in the equation:

$$\Phi^{t+1} = \Phi^t - \alpha \frac{1}{N} \frac{\partial C(X, \Phi^t)}{\partial \Phi^t} \tag{7}$$

where Φ^t denotes the parameters of the neural network at iteration t in the gradient descent, and X is a vector consisting of N randomly picked input-output pairs from the dataset. Other state-of-the-art optimizers include Adam and Root Mean Square Propagation (RMSprop).

2.3 Convolutional neural networks

When it comes to image analysis, basic neural networks tend to overfit: the fully connected structure of the network results in too many tunable parameters with high dimensional input.

Convolutional neural networks (CNN) are an advancement in the space of neural networks, architecture especially suited for tasks with high dimensional input like images. It was first presented in the 1995 whitepaper ”Convolutional Networks for Images, Speech, and Time-Series” by Le Cunn and Bengio [11], and has since been one of the most used architectures in the field of deep learning. The original CNN architecture is shown in figure 5.

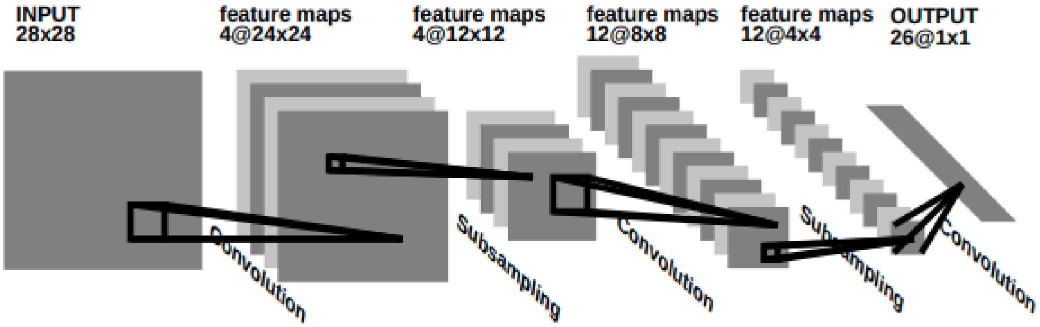
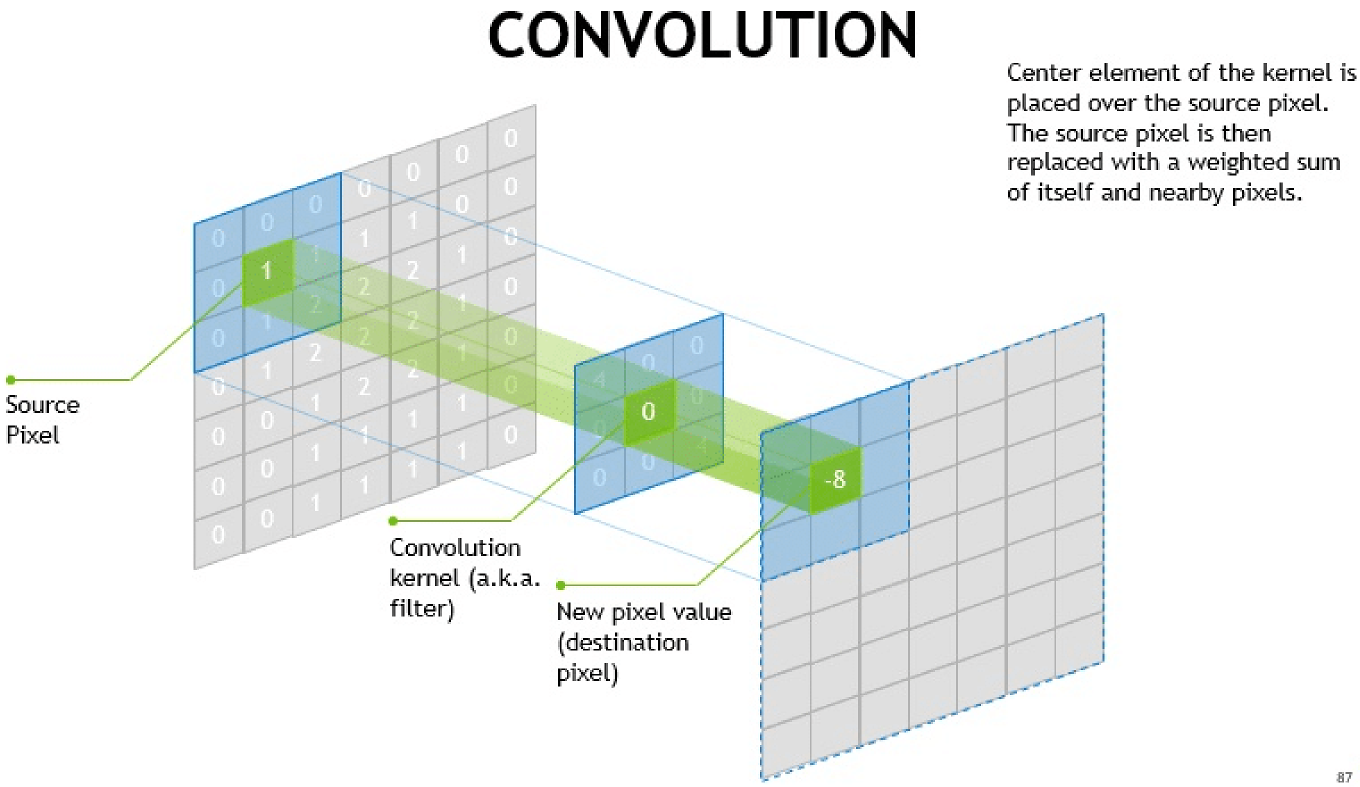


Figure 5: Convolutional Neural Network for image processing, e.g., handwriting recognition [11]

The main idea with CNNs is to come up with learnable filters that convolve over the original images, extracting features useful for the classification. There can be multiple filters per layer, resulting in multiple convolved images as input for the next layer. These filters can learn to extract different kinds of features through convolution, as described in figure 6. This automatic feature extraction makes all the three main stages of a classification problem (feature extraction, feature selection, and classification) directly learnable from the data for a CNN.



CONVOLUTION

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

Source Pixel

Convolution kernel (a.k.a. filter)

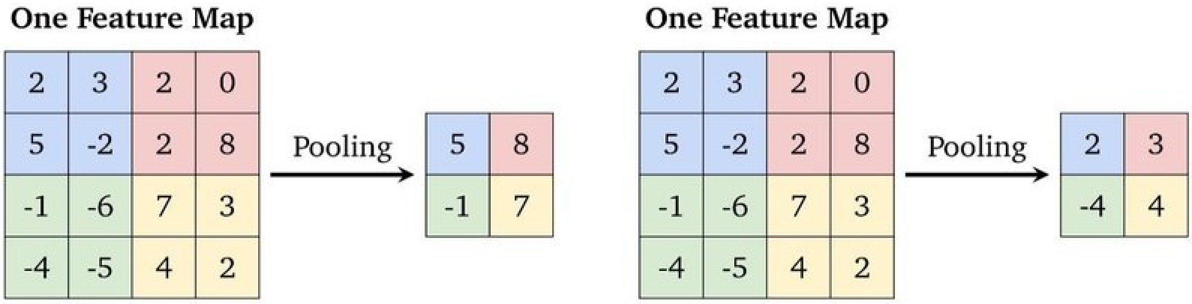
New pixel value (destination pixel)

Figure 6: The convolutional filter is the key element in a CNN. It acts as a mathematical filter that helps computers find edges of images, dark and light areas, colors, and other details, such as height, width and depth. [12]

After the convolution, the pixel values of these convolved images, called feature maps, are rescaled with a chosen nonlinear activation function. Then, the feature maps are downsampled regionally either by averaging the elements or by taking the maximum element (see figure 7). This downsampling reduces the dimensionality while keeping the most influential features. It also makes the network invariant to small rotations of the input, resulting in better generalization.

This process is then repeated multiple times depending on the depth of the CNN. Finally, the last set of feature maps are flattened to produce a one-dimensional feature vector which is then fed to a basic neural network producing the output of the network. The loss of the network is calculated the same way as with the basic neural network, and the principles of forward- and backpropagation stay the same (with the addition of taking the derivative of the pooling and the convolution operations).

It has been found that the initial convolutional layers extract universal features, like edges and curves, and the final convolutional layers extract features more specific to the classification task like faces. This task invariance in the earlier layers of a CNN has made it a good candidate for a transfer learning model.



(a) Max-Pooling

(b) Average-Pooling

Figure 7: Reduction in the feature map size due to (a) max-pooling and (b) average-pooling. [13]

2.4 Transfer learning

Transfer learning is an idea that we can use models trained on a similar task and get good results by just finetuning the model to our particular task. This approach aims to avoid the problems of data acquisition and computational cost of training a model from scratch (i.e. random initialization of the parameters).

CNN architecture is suitable for this method since the earlier convolutional layers have been found to extract universal shapes like edges and curves. This means that we can focus the training on the last, task-specific neural network and still attain good results with a reduced computational cost for the training and with fewer samples available.

For example, in [14], it was discovered that transfer learning is possible between domains of gesture recognition from Electromyographic (EMG) signals and the mental state recognition from Electroencephalographic (EEG) brainwaves. The experiments found that the accuracy of the neural network was improved through prior exposure to another domain both at the first epoch (prior to any learning on the current domain) and with the final achieved performance compared to random parameter initialization.

2.5 Visualization of the classification result of the CNN

There are a lot of benefits for visualizing the classification result of the CNN. Not only does it increase the explainability of the decision-making of the CNN to a human user, but it can also help to choose a better model out of two even if their accuracies would be the same.

We could, for example, observe that model A picks up on some feature of the brain on an MRI scan, but model B picks something from the background which could be considered a bug in the model B or in training data. Proper visualization of the classification result could thus reveal faults in the model or in the collected data, which might be overlooked if only concentrating on the classification accuracy.

This can help us better predict how the model would perform with unseen data.

Visualization could also reveal new important features, on which further research could be based upon. Thus proper visualization has the ability to better align machine learning and human learning and to make machine learning more beneficial for its users.

There are some existing studies that compare different visualization methods of the CNN decision making in the MRI classification setting [15], [16]. Two key techniques are gradient-based techniques and occlusion-based techniques, both trying to highlight which parts of the input contribute most to the decision-making.

In the occlusion-based techniques, part of the input image is occluded (pixels are set to zero), and the classification result of the occluded image is compared to the classification result of the non-occluded image. Changes in the class activation in the output of the network are set to be the importance of the occluded region in the image for that class. This process is described in figure 8.

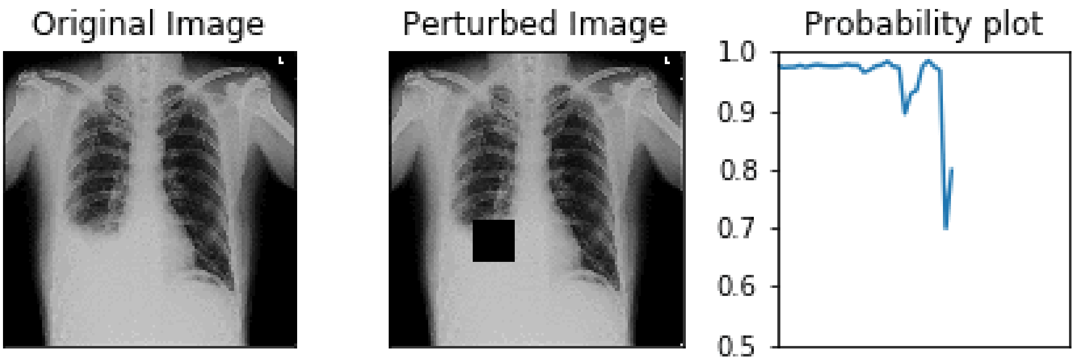


Figure 8: In occlusion-based techniques, part of the image is occluded and the effect on the class probability is observed. With this chest X-ray image of a patient diagnosed with pleural effusion, the probability plot is updated as the occlusion patch shifts over the image. We can observe that occluding the current region makes the pleural effusion class probability drop drastically, i.e., the region is important for that class. [17]

Gradient-based techniques backpropagate the class activation back to the layers of the network. With the basic method, referred to as Gradient-weighted Class Activation Mapping (Grad-CAM), the last convolutional layer is the suggested layer to start with, since it offers a good balance between resolution and class discrimination. Gradients of the earlier layers would offer better resolution, but they do not discriminate features between the classes as well [18]. The Grad-CAM method will be discussed in further detail in chapter 3.3 in data and methods section.

As per MATLAB's documentation [20], occlusion-based techniques and gradient-based techniques usually return similar kinds of results (see figure 9), although they work in different ways. Usually, you can compute the Grad-CAM map more efficiently, with just a single forward and backward pass and without tuning any parameters. However, Grad-CAM usually has a lower spatial resolution than an

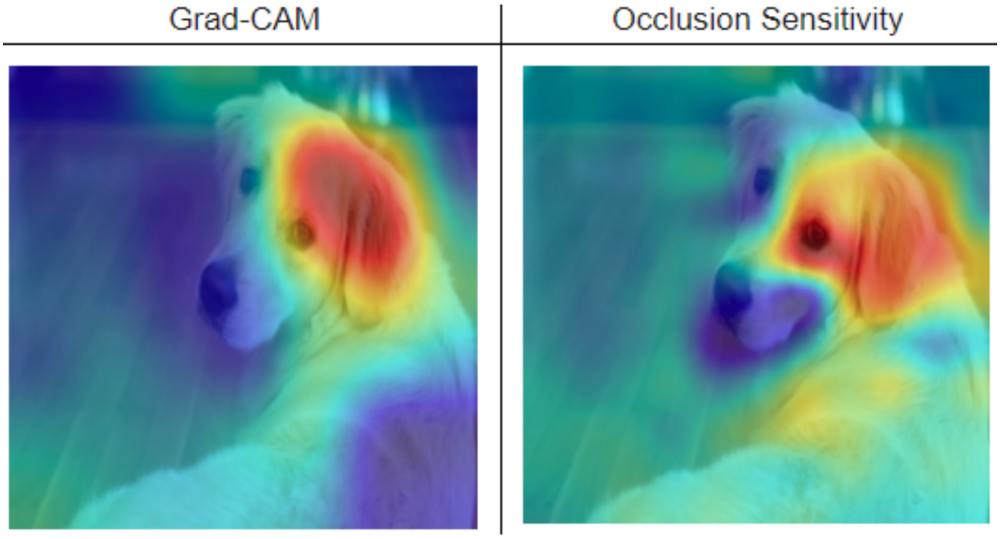


Figure 9: Occlusion-based and gradient-based techniques usually return similar kinds of results. Here the results of both methods are visualized as heatmaps, where warmer colour means higher impact on the class activation. [19]

occlusion map and so it can miss fine details. The resolution of the Grad-CAM map is the resolution of the chosen feature map layer. This is in contrast with the occlusion-based methods, where the occlusion size is a parameter to be tuned.

In the empirical section of this thesis, Grad-CAM is the chosen visualization method since the computational resources available were limited and the method is computationally efficient. Grad-CAM is also seen as a natural extension to the backpropagation already being done in the training phase of the CNN, and so possibly being better suited to explain the decision making of the network.

3 Data and methods

3.1 Data

The data used is from a Kaggle repository *Brain MRI Images for Brain Tumor Detection* owned by Navoneel Chakrabarty [21]. The images are 2D and they are collected from Google Images by the author. The images are a mix of T1 and T2 pulse-sequenced MRI images. The labels of the images are "no" and "yes" answering the question is there a tumor visible in the image. There are 253 images combined, including 98 images with the label "no" and 155 images with the label "yes".

Kaggle, which is a subsidiary of Google LLC, is an online community to find and publish datasets and models and to collaborate with other data scientists and machine learning practitioners. Kaggle also offers machine learning competitions, from where users can then easily see some achieved results on particular datasets and compare their own approach and results with them. The top model listed in the used repository is the Brain Tumor Detection v1.0 || CNN, VGG-16 by Ruslan Klymintiev. In the posted article about the model, [22], accuracies of 88% and 80% were listed for validation set and test set respectively.

3.2 Transfer learning model: VGG-16

The chosen deep learning architecture is a 16 layer deep CNN named VGG-16 [23] (the name referring to the number of layers and the name of the group who build it, visual geometry group). The pretrained version is trained on more than a million images from the ImageNet database, containing images from more than 20,000 categories such as "balloon" or "strawberry".

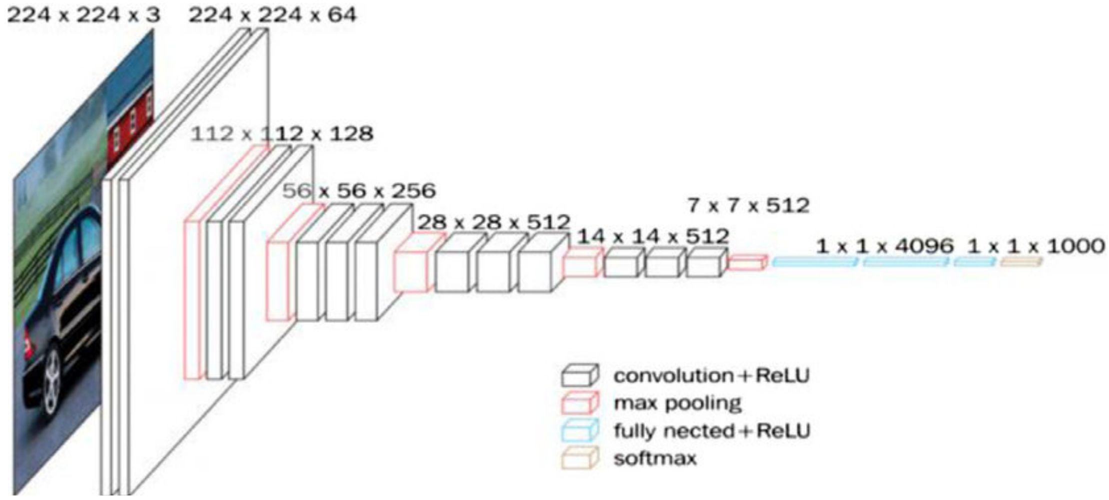


Figure 10: Structure of the VGG-16. In this thesis implementation, the structure was modified to accept grayscale input by modifying the first convolutional layer. The last 3 fully connected layers were also replaced with only one fully connected layer with two output nodes corresponding to the two classes available in the dataset.

The original architecture, depicted in the figure 10, consists of 13 convolutional layers and 3 fully connected layers, which are divided into blocks the following way: 2 contiguous blocks of 2 convolutional layers followed by a max-pooling, then 3 contiguous blocks of 3 convolutional layers followed by max-pooling and at last 3 fully connected layers. The last 3 fully connected layers depend on the specifics of the task. Both the pretrained version and the nontrained version of the network were tested on the dataset, to see if transfer learning would bring any benefits between domains of natural images and medical images.

3.3 Visualization method: Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a method to visualize the classification result of the CNN. It was first presented in 2016 in the paper "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization" by Selvaraju et al. [18]. In the Grad-CAM, the class activation (the input to the activation function at the output layer) is backpropagated to the final feature map layer to get the weights for each feature map. The weights are calculated by summing the individual pixel gradient values of a feature map:

$$\alpha_k^c = \sum_{x,y} \frac{\partial y_c}{\partial f_k(x,y)}, \tag{8}$$

where $f_k(x,y)$ is the pixel value at x,y-location in kth feature map in the final feature map layer. The weights α_k^c measure the importance of the corresponding feature maps for the class activation y_c in the output layer of the network (before applying softmax). Weighted feature maps are then summed and activated using ReLU in order to only leave active the regions having a positive effect on the class activation:

$$G(x,y) = ReLU(\sum_k \alpha_k^c f_k(x,y)). \tag{9}$$

The resulting class-activation map $G(x,y)$ is then resized to the size of the input image using the bilinear interpolation method. The whole method is visualized in figure 11.

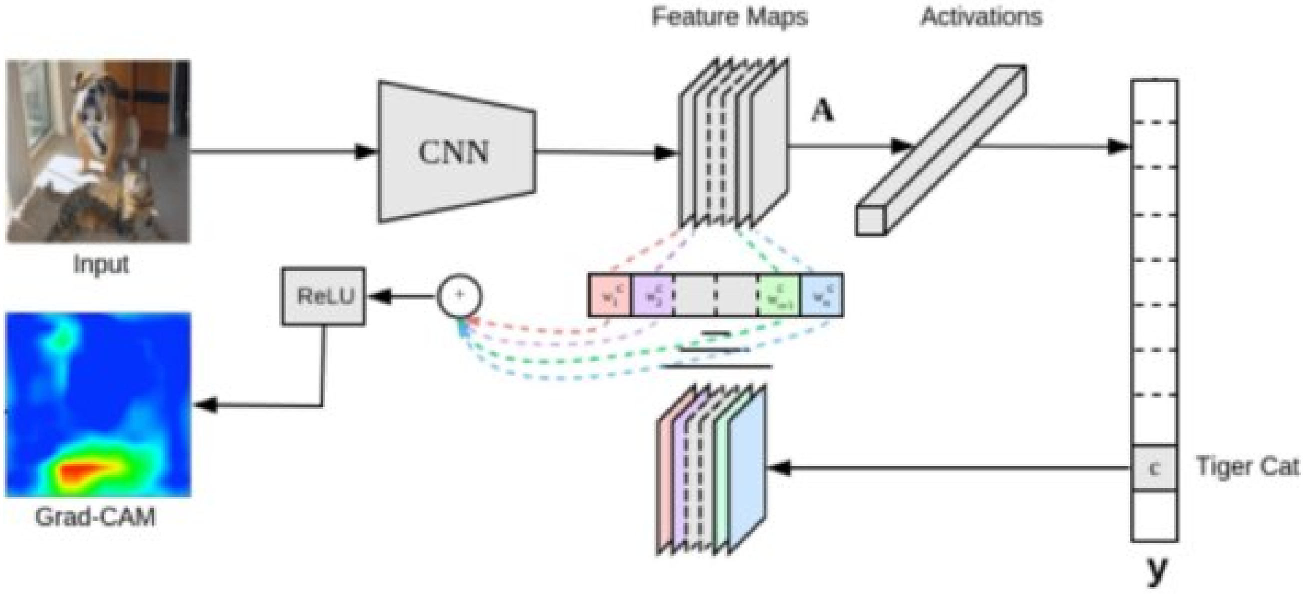


Figure 11: Structure of the Grad-CAM visualization method. [18]

3.4 Classification and visualization pipeline

The data was split into training, validation, and test sets by an 80%-10%-10% split. The samples were drawn so that the first 80% of the samples of both labels were to go to the training set, the next 10% to the validation set, and the last 10% to the testing set. Order is the directory structure of the downloaded files. This resulted in training, validation, and test sets having 202, 26, and 25 samples respectively, all having the class ratio same as in the whole data (40% "no" labels and 60% "yes" labels).

The VGG-16 model, originally trained with RGB images, was adjusted to work with grayscale images by averaging the weights in the channel dimension in the first convolutional layer. The image input layer was also replaced to have only one color channel. Normalization of the input was set as zero-center, i.e., the mean of the pixel values of all the samples was subtracted from each sample.

The images were resized to the input size of the network using the bilinear interpolation. Data augmentation was applied to improve model generalization to unseen data [24]. Data augmentation included random -15 to +15 pixels translation on x- and y-axis, -15 to +15% random scale change, -15° to +15° random rotation and random reflection along x-axis.

The optimizer used in the training was stochastic gradient descent with a momentum of 0.9 (SGDM). The learning rate was set to 0.0001 and the minibatch size was 20 (i.e., the gradient estimate was calculated based on 20 randomly picked samples out of the whole dataset). The maximum amount of epochs was set at 100 and the validation patience was set as 10 epochs (1 epoch corresponding to going through the whole dataset once with the minibatches). This means that the training was stopped if validation loss did not improve for 10 epochs in a row or the training had already lasted for 100 epochs. Grad-CAM was used for the visualization of the decision making. The whole classification and visualization pipeline implementation is available in MATLAB code at: https://github.com/juuso-oskari/CNN-MRI-visualization.

4 Results

4.1 Evaluation metrics

True positive (TP) means a correctly classified image having a brain tumor (i.e. having label "yes"), where as false positive (FP) means falsely classifying image to contain a brain tumor when the correct label is "no". True negative (TN) means correctly classifying image to not contain a brain tumor (i.e. having label "no"), where as false negative (FN) means falsely classifying image to not contain a brain tumor when the correct label is "yes". Combining these terms, I used the following metrics to describe the model classification performance:

- Accuracy $= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$
- Specificity $= \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%$
- Sensitivity $= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%$
- Precision $= \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%$

In addition to the accuracy, the average cross-entropy loss was used to describe the training performance (referred to as loss in further reading). In Grad-CAM visualizations, the following things were evaluated: the ability to locate the tumor region, boundary information (i.e., discrimination between healthy tissue and tumor tissue at the boundary), confusion (i.e., are there non-tumor regions highlighted), ability to infer generalization of the model from the visualizations and do the visualizations make the misclassifications more reasonable.

4.2 Effect of transfer learning on the training performance

The training performance was better with the transfer learning model compared to the model with random initialization (see figure 12). As an example, at around the 10th epoch, the validation accuracy and the validation loss had already reached 94% and 0.2 with the transfer learning model, whereas with the model with random initialization they were still at 83% and 0.5. The training of the transfer learning model was stopped at epoch 55 due to the validation loss converging (i.e., validation loss did not improve for 10 epochs in a row). The training of the model with random initialization ran for a maximum of 100 epochs. The final achieved validation accuracy, 96.15%, was the same for both models (although the transfer learning model reached 100% validation accuracy on several occasions during training), but the final validation loss was better with the pretrained model: 0.069 versus 0.1507. The pretrained model also had much better performance on the test dataset which was not used in the training phase: the pretrained model achieved an accuracy of 88%, whereas the model with random initialization achieved an accuracy of 76%.

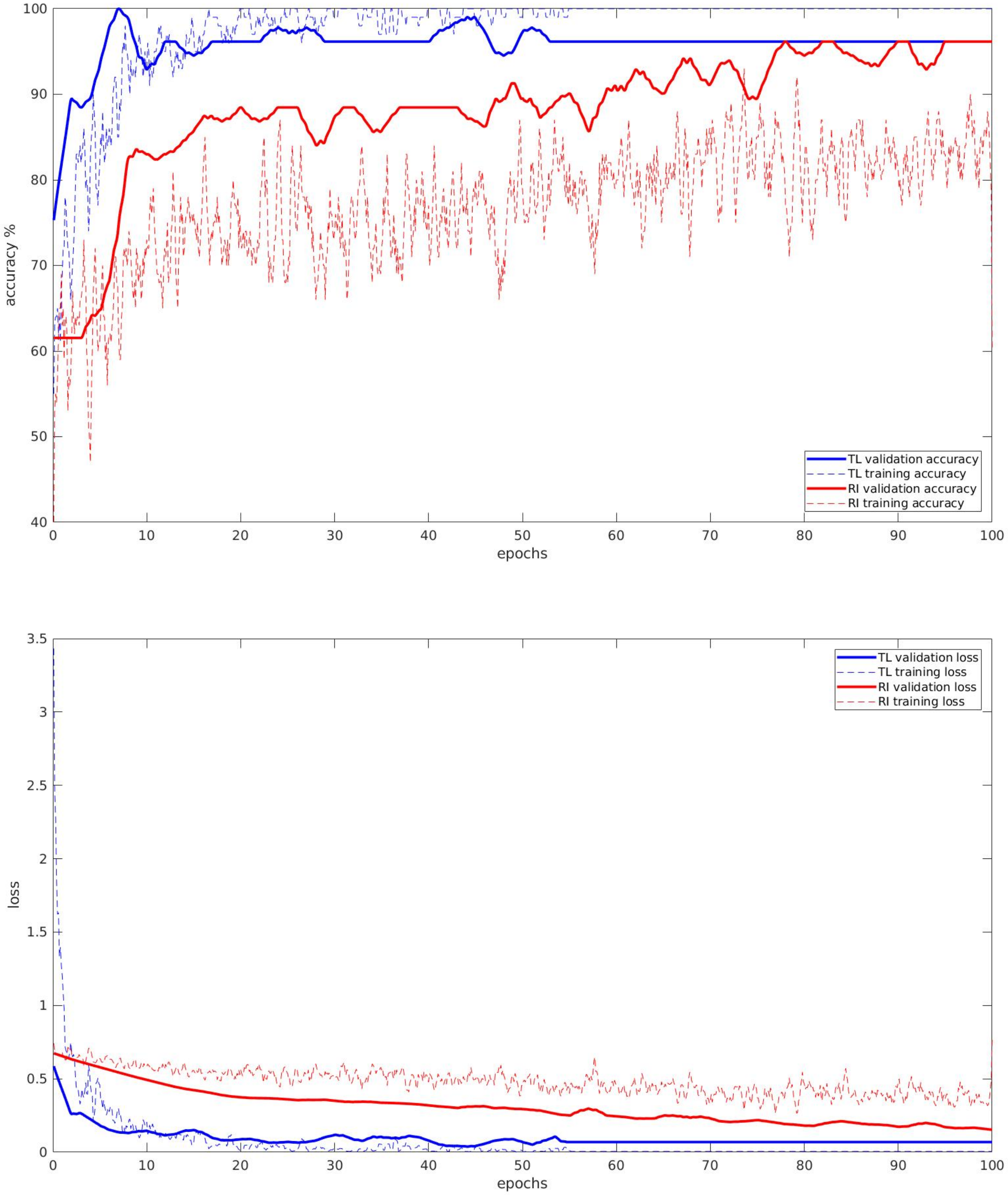


Figure 12: Training performance of the transfer learning (TL) model and the model with random initialization (RI). Training was stopped at epoch 55 for the TL model due to converging. RI model was trained for the maximum of 100 epochs.

4.3 Classification results

The transfer learning model achieved 88.0% accuracy on the test set. From the confusion matrix (shown in figure 13), we can observe that the specificity was 90.0%, sensitivity was 86.7%, and precision 92.9%. The training was stopped early at epoch 55 due to the validation loss not improving for the last 10 epochs. The achieved accuracies and losses for the training and the validation sets were 100% and 96.15%, and 0.005 and 0.069, respectively (see table 1).

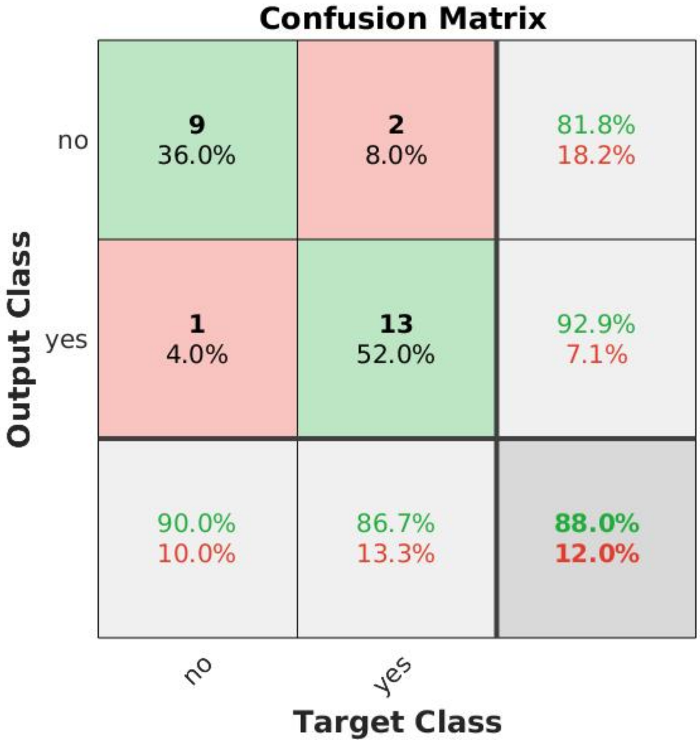


Figure 13: Testing set confusion matrix of the pretrained model. Target output labels are at the horizontal axis, and predicted output labels are at the vertical axis. Diagonal elements represent correct predictions.

In contrast, the model with random initialization achieved 76.0% accuracy on the test set. From the confusion matrix (shown in figure 14), we can observe that the specificity was 50.0%, sensitivity was 93.3% and precision was 73.7%. The training lasted for a maximum of 100 epochs. The achieved accuracies and losses for training and validation sets were 85% and 96.15%, and 0.767 and 0.151, respectively. For comparison of the results between the transfer learning model and the model with random initialization, see table 1.

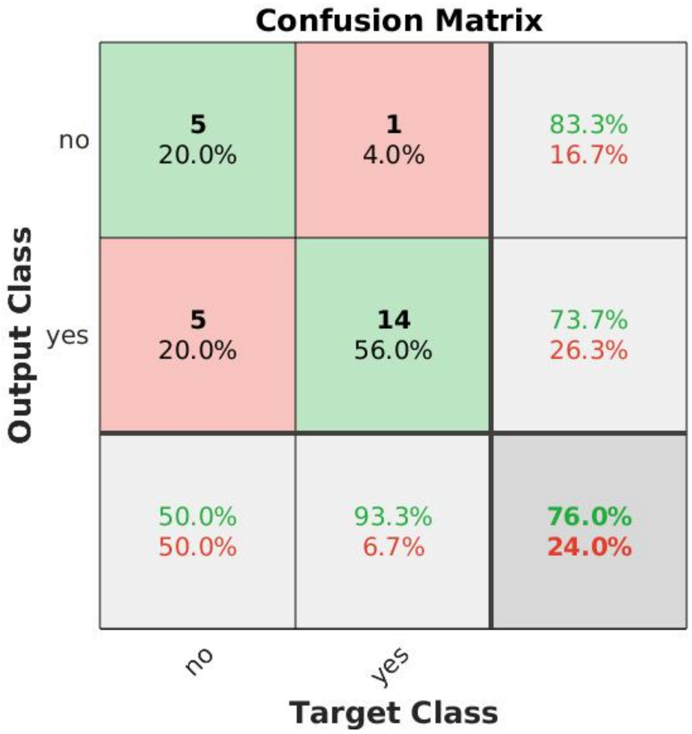


Figure 14: Test set confusion matrix of the model with random initialization. Target output labels are at the horizontal axis, and predicted output labels are at the vertical axis. Diagonal elements represent correct predictions.

Model	Val. acc.	Val. loss	Test acc.	Test spec.	Test sens.	Test prec.
Transfer learning	96.15%	0.069	88.0%	90.0%	86.7%	92.9%
Random initialization	96.15%	0.151	76%	50.0%	93.3%	73.7%

Table 1: Table of classification results. The abbrevations are as follows: val. means validation, acc. accuracy, spec. specificity, sens. sensitivity, prec. precision.

4.4 Visualization results

For most of the true positive images, the Grad-CAM visualization worked adequately. In table 2 are the visualization results for 5 randomly selected true positive samples out of the test set. These visualizations give an indication of the region the network found important for its decision-making.

The resolution of these regions, however, is quite poor and the boundary information seems not reliable (i.e. the heatmap does not clearly outline the boundary of healthy tissue and tumor tissue). This is due to the low resolution of the final feature map layer. The VGG-16 results in 14x14 feature maps in the final feature map layer.

The visualization also appeared to be wrong with the image of a deformed brain with extra-large ventricles (image 4 from the top in table 2), even though the network was able to classify it correctly. This could mean that the dataset did not contain many deformed, but otherwise healthy brains (i.e. deformation of the brain was a reliable indicator of the image belonging to a class "yes").

The region of importance also seemed a little shifted compared to the tumor region in the underlying original image. This is most likely due to the loss in spatial localization due to max-pooling done to the input image in the VGG-16. For example, max-pooling with a 3x3 filter makes one lose the information which of the original 9 pixels is the maximum element. Consequent max-pooling operations make the spatial information loss even worse.

In table 3, the visualizations were also performed for 5 randomly selected true positive samples out of the validation set for both the transfer learning model and the model with random initialization. This was done to see if one is able to infer the better generalization ability of the transfer learning model from the visualizations, even though the models achieved exactly the same accuracy on the validation set.

The answer, however, remains unclear. In some of the visualizations (images 1, 2, and 5 from the top in the table 3) the transfer learning model is better at locating the tumor tissue, which could give a hint for better generalization ability. But with the rest of the images, both of the models seem confused about the location of the tumor tissue (even though they were able to classify the images correctly).

These confusing images further reassure the point that with a limited amount of data available, the network can make non-generalizable conclusions, even though the accuracy would be good. For example, with the model with the random initialization, it is easy to see that the inclusion of soft tissue in the image could produce a false positive classification result since soft tissue from eye sockets is also highlighted in images 3 and 4 from the top in the table 3.

The visualizations for the misclassified images (shown in table 4) offered no clear explanation for why the images were misclassified. The first image from the top has the upper part of the brain highlighted where there is also soft tissue from eye sockets, and the rest of the images seem to have some contrast-issues confusing the network. But these are only speculative interpretations and do not make the misclassifications appear reasonable.

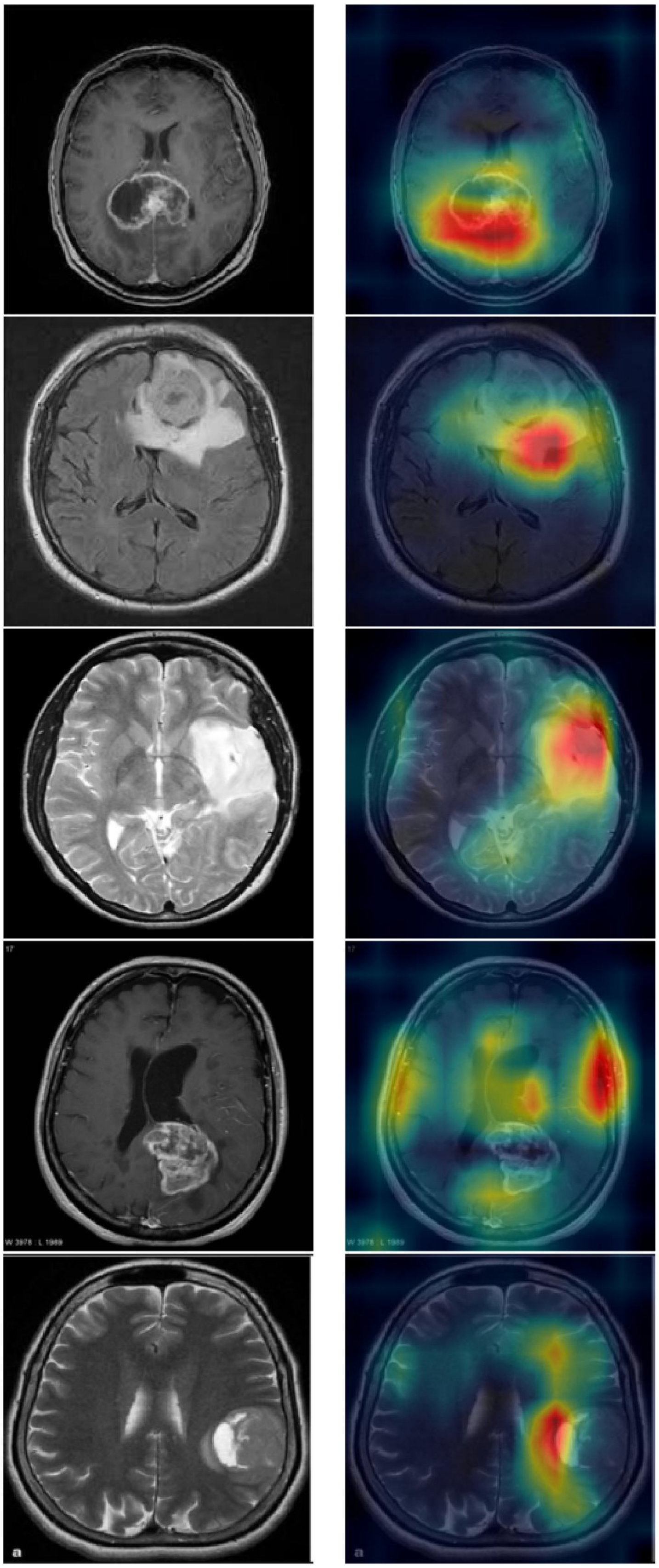


Table 2: The Grad-CAM visualizations performed on 5 randomly selected true positive samples out of the test set. In the visualizations warmer color means that the region has a higher impact on the class activation based on the gradient analysis.

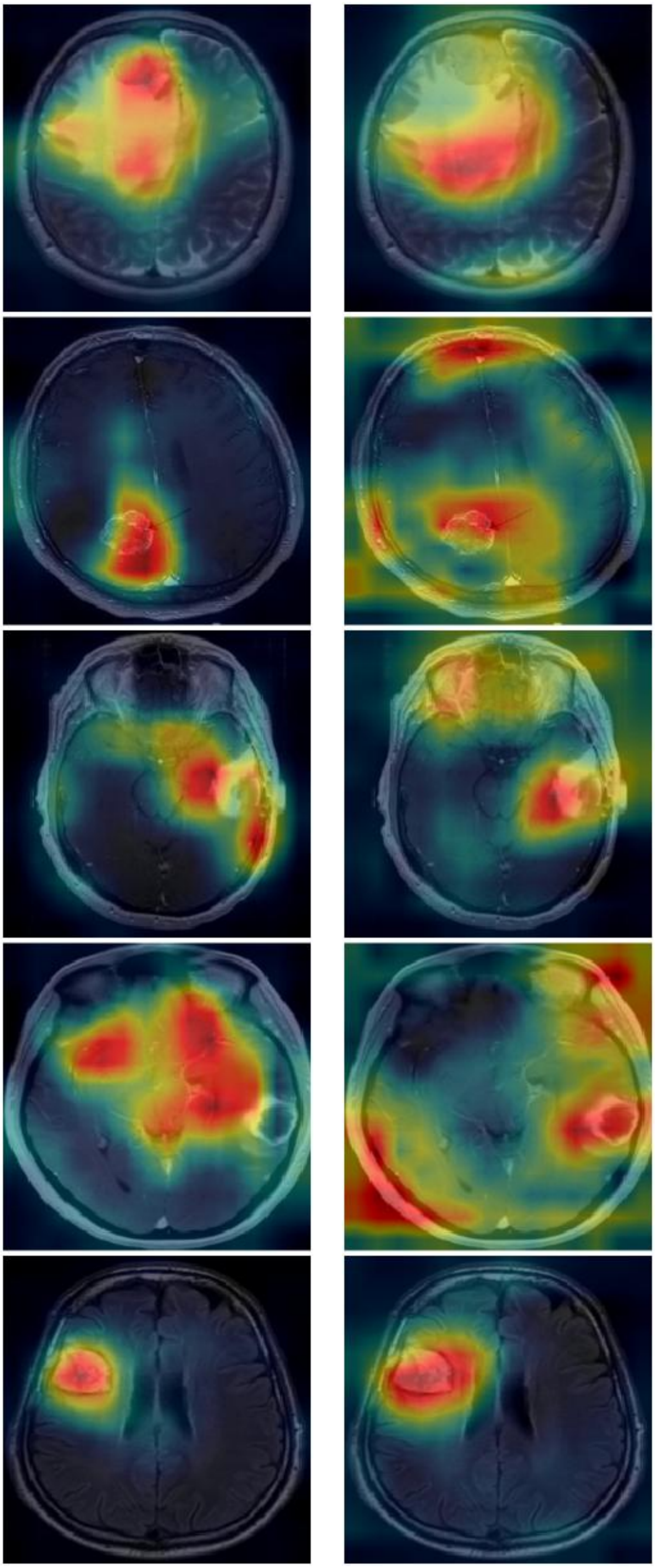


Table 3: The Grad-CAM visualizations performed on 5 randomly selected true positive samples out of the validation set for both the transfer learning model (images on the left) and the model with random initialization (images on the right). In the visualizations warmer color means that the region has a higher impact on the class activation based on the gradient analysis.

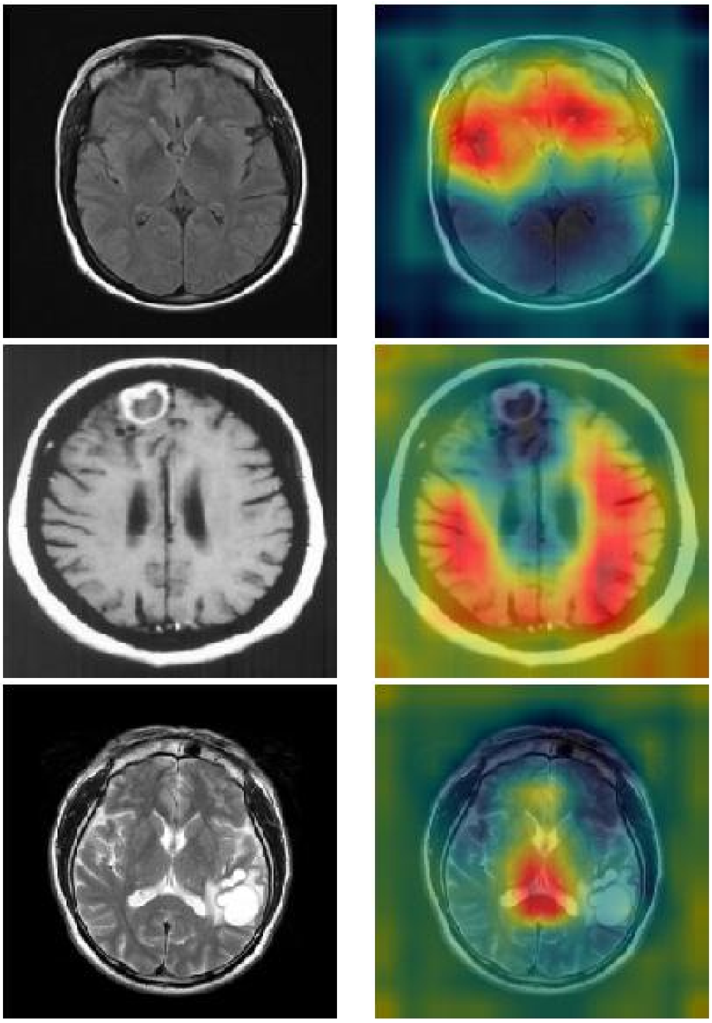


Table 4: The Grad-CAM visualizations performed on all the 3 misclassified images. From the top, the first image had a true label "no", but the predicted label was "yes" with a certainty of 0.6. The second image had a true label "yes", but the predicted label was "no" with a certainty of 0.97. The third image also had a true label "yes", but the predicted label was "no" with a certainty of 0.96. In the visualizations warmer color means that the region has a higher impact on the class activation based on the gradient analysis.

5 Conclusions

The thesis set out to investigate the suitability of CNNs in the diagnosis of MRI brain scan. The most important advantage with CNNs compared to other machine learning methods was found to be the adaptive feature extraction done in the convolutional phase of the network. This makes it possible to use the raw MRI image as an input and the optimization of all the steps of the classification task based on the training data. The thesis also identified and tested solutions for the two key problems encountered when introducing CNNs into the field: data scarcity and explainability of the decision-making.

Transfer learning was found to be a viable solution to alleviate the problems of data scarcity. The training performance and the classification performance were better with the transfer learning model compared to the model with random initialization. The training and validation losses converged earlier (see figure 12) and the reached accuracies and losses with the test set were better with the transfer learning model (see table 1). The test set classification performance of the transfer learning model was comparable to that of the top model listed on the dataset repository page [22].

The Grad-CAM visualization method for the final feature map layer proved to be a good first step to better understand the CNN decision-making. The visualizations offered indication for the region that the CNN based its classification on (i.e., where it thought the tumor was located). The visualizations also offered some indication for the generalization ability of the model, since they were somewhat better with the transfer learning model compared to the model with random initialization.

However, the visualizations had poor resolution overall and for misclassifications, the visualizations did not explain why the misclassifications were made. To offer better resolution and better boundary information, considerations for the network architecture have to be made. The VGG-16 used in this work results in 14x14 feature maps after the final convolutional layer. Most of the deeper and better performing models result in even smaller resolutions, so the resolution issues would most likely only increase.

One approach to improve the detail level of the visualization method would be to some way include the earlier layers offering better resolution. The original paper [18] already introduced the idea to take the gradient of the class activation with respect to the input image, in order to bring out finer details in combination with the low-resolution maps produced by the original Grad-CAM approach. However, as a well-trained deep network is not strongly dependent on the exact value of specific pixels, it is not clear how much insight these extensions can provide into the model [25].

6 References

References

[1] Yu, L. & Chen, H. & Dou, Q. & Qin, J. & Heng, P. A. *Automated melanoma recognition in dermoscopy images via very deep residual networks.* IEEE Transactions on Medical Imaging. Vol. 36:4. 2017. Pp. 994-1004. ISSN 1558-254X. DOI:10.1109/TMI.2016.2642839.

[2] Ebrahimi, A. & Luo, S. & Chiong, R. *Introducing transfer learning to 3d resnet-18 for alzheimer's disease detection on mri images.* 35th International Conference on Image and Vision Computing New Zealand (IVCNZ). 2020. Pp. 1-6. ISSN 2151-2205. DOI: 10.1109/IVCNZ51579.2020.9290616.

[3] Smith-Bindman, R. & Kwan, M. L. & Marlow, E. C & Theis, M. K & Bolch, W. & Cheng, S. Y. & Bowles, E. J. A & Duncan, J. R. & Greenlee, R. T. & Kushi, L. H & Pole, J.D & Rahm, A.K. & Stout, N. K. & Weinmann, S. & Miglioretti, D. L. *Trends in Use of Medical Imaging in US Health Care Systems and in Ontario, Canada, 2000-2016.* JAMA. Vol. 322. 2019. Pp. 843-856. DOI:10.1001/jama.2019.11456.

[4] Greenspan, H. & van Ginneken, B. & Summers, R. M. *Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique.* IEEE Transactions on Medical Imaging. Vol. 35:5. 2016. Pp. 1153-1159. DOI: 10.1109/TMI.2016.2553401.

[5] Litjens, G. & Kooi, T. & Bejnordi, B. E. & Setio, A. A. A. & Ciompi, F. & Ghafoorian, M. & van der Laak, J. & van Ginneken, B. & Sánchez, C. *A survey on deep learning in medical image analysis.* Medical Image Analysis. Vol. 42. 2017. Pp. 60-88. ISSN 1361-8415. Available: https://doi.org/10.1016/j.media.2017.07.005.

[6] Krizhevsky, A. & Sutskever, I. & Hinton, G. E. *Imagenet classification with deep convolutional neural networks.* Communications of the ACM. Vol. 60:6. 2017. ISSN 0001-0782. DOI:10.1145/3065386.

[7] Paperswithcode. *Browse State-of-the-Art.* [Accessed at 5.5.2021]. Available: https://paperswithcode.com/sota.

[8] Rosenblatt, F. *The perceptron: A probabilistic model for information storage and organization in the brain.* Psychological Review. Vol 65:6. 1958. Pp. 386-408. Available: https://doi.org/10.1037/h0042519.

[9] Hornik, K. *Approximation capabilities of multilayer feedforward networks.* Neural Networks. Vol. 4:2. 1991. Pp. 251-257. ISSN 0893-6080. Available: https://www.sciencedirect.com/science/article/pii/089360809190009T.

[10] Goodfellow, I. & Bengio, Y. & Courville, A. *Deep Learning.* Cambridge, MA, USA: MIT Press. 2004. Pp. 200. Available: http://www.deeplearningbook.org.

[11] LeCun, Y. & Bengio, Y. *Convolutional Networks for Images, Speech, and Time Series.* Cambridge, MA, USA: MIT Press. 1998. Pp. 255-258. ISBN 0262511029.

[12] Scott, M. *What's the difference between a cnn and an rnn?.* 2018. [Accessed at 25.3.2021]. Available: https://blogs.nvidia.com/blog/2018/09/05/whats-the-difference-between-a-cnn-and-an-rnn/

[13] Gong, W. & Chen, H. & Zhang, Z. & Zhang, M. & Wang, R. & Guan, C. & Wang, Q. *A Novel Deep Learning Method for Intelligent Fault Diagnosis of Rotating Machinery Based on Improved CNN-SVM and Multichannel Data Fusion.* Sensors. Vol. 19:7. 2019. ISSN 1424-8220. Available: https://doi.org/10.3390/s19071693.

[14] Bird, J. J. & Kobylarz, J. & Faria, D. R. & Ekárt, A. & Ribeiro, E. P. *Cross-domain mlp and cnn transfer learning for biological signal processing: Eeg and emg.* IEEE Access. Vol. 8. 2020. Pp. 54789-54801. DOI: 10.1109/ACCESS.2020.2979074.

[15] Yang, C. & Rangarajan, A. & Ranka, S. *Visual explanations from deep 3d convolutional neural networks for alzheimer's disease classification.* AMIA Annu Symp Proc. 2018. Pp. 1571-1580. Available: https://arxiv.org/abs/1803.02544.

[16] Kim, I. & Rajaraman, S. & Antani, S. *Visual interpretations of convolutional neural network predictions in classifying medical image modalities.* Diagnostics. Vol. 9:2. 2019. Pp. 1-38. DOI: 10.3390/diagnostics9020038.

[17] Ghosh, R. & Jain, S. & TLD, M. *Visualizing deep learning networks - part I.* 2017. [Accessed 16.4.2021]. Available: https://blog.qure.ai/notes/visualizing_deep_learning.

[18] Selvaraju, R. R. & Das, A. & Vedantam, R. & Cogswell, M. & Parikh, D. & Batra, D. *Grad-CAM: Why did you say that? Visual explanations from deep networks via gradient-based localization.* CoRR. Vol. abs. 2016. 1610.02391. Available: https://arxiv.org/abs/1610.02391.

[19] Pingel, J. *Understanding and using deep learning networks.* 2020. [Accessed 21.4.2021]. Available: https://blogs.mathworks.com/deep-learning/2020/09/30/new-deep-learning-examples/.

[20] Mathworks. *Understanding and using deep learning networks (R2021a).* [Accessed 8.5.2021]. Available: https://se.mathworks.com/help/deeplearning/ug/understand-network-predictions-using-occlusion.html.

[21] Chakrabarty, N. *Brain mri images for brain tumor detection.* 2019. [Accessed at 2.3.2021]. Available: https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection.

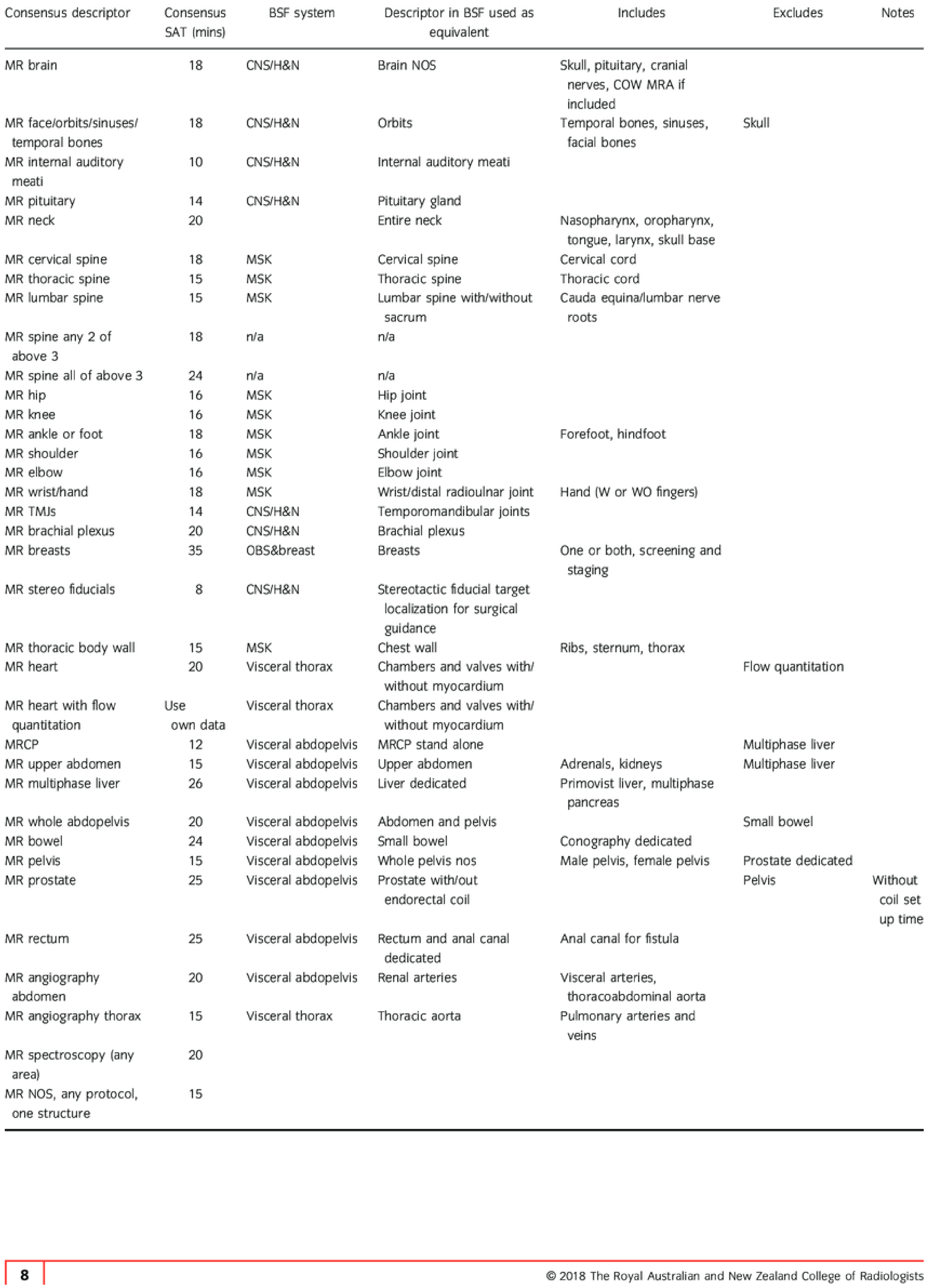
[22] Klymentiev, R. *Brain tumor detection v1.0 || cnnm vgg-16 [online].* 2019. [Accessed at 2.3.2021]. Available: https://www.kaggle.com/ruslankl/brain-tumor-detection-v1-0-cnn-vgg-16.

[23] Simonyan, K. & Zisserman, A. *Very deep convolutional networks for large-scale image recognition.* 2014. Available: https://arxiv.org/abs/1409.1556.

[24] Shorten, C. & Kohsgoftaar, T. *A survey on image data augmentation for deep learning.* Journal of Big Data. Vol. 6. 2019. Pp. 1-48. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0.

[25] Adebayo, J. & Gilmer, J. & Muelly, M. & Goodfellow, I. & Hardt, M. & Kim, B. *Sanity checks for saliency maps.* Advances in Neural Information Processing Systems. Vol. 31. 2018. Available: https://proceedings.neurips.cc/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf.

A Appendix



Consensus descriptor	Consensus SAT (mins)	BSF system	Descriptor in BSF used as equivalent	Includes	Excludes	Notes
MR brain	18	CNS/H&N	Brain NOS	Skull, pituitary, cranial nerves, COW MRA if included		
MR face/orbits/sinuses/ temporal bones	18	CNS/H&N	Orbits	Temporal bones, sinuses, facial bones	Skull	
MR internal auditory meati	10	CNS/H&N	Internal auditory meati			
MR pituitary	14	CNS/H&N	Pituitary gland			
MR neck	20		Entire neck	Nasopharynx, oropharynx, tongue, larynx, skull base		
MR cervical spine	18	MSK	Cervical spine	Cervical cord		
MR thoracic spine	15	MSK	Thoracic spine	Thoracic cord		
MR lumbar spine	15	MSK	Lumbar spine with/without sacrum	Cauda equina/lumbar nerve roots		
MR spine any 2 of above 3	18	n/a	n/a			
MR spine all of above 3	24	n/a	n/a			
MR hip	16	MSK	Hip joint			
MR knee	16	MSK	Knee joint			
MR ankle or foot	18	MSK	Ankle joint	Forefoot, hindfoot		
MR shoulder	16	MSK	Shoulder joint			
MR elbow	16	MSK	Elbow joint			
MR wrist/hand	18	MSK	Wrist/distal radioulnar joint	Hand (W or WO fingers)		
MR TMJs	14	CNS/H&N	Temporomandibular joints			
MR brachial plexus	20	CNS/H&N	Brachial plexus			
MR breasts	35	OBS&breast	Breasts	One or both, screening and staging		
MR stereo fiducials	8	CNS/H&N	Stereotactic fiducial target localization for surgical guidance			
MR thoracic body wall	15	MSK	Chest wall	Ribs, sternum, thorax		
MR heart	20	Visceral thorax	Chambers and valves with/ without myocardium		Flow quantitation	
MR heart with flow quantitation	Use own data	Visceral thorax	Chambers and valves with/ without myocardium			
MRCP	12	Visceral abdopelvis	MRCP stand alone		Multiphase liver	
MR upper abdomen	15	Visceral abdopelvis	Upper abdomen	Adrenals, kidneys	Multiphase liver	
MR multiphase liver	26	Visceral abdopelvis	Liver dedicated	Primovist liver, multiphase pancreas		
MR whole abdopelvis	20	Visceral abdopelvis	Abdomen and pelvis		Small bowel	
MR bowel	24	Visceral abdopelvis	Small bowel	Conography dedicated		
MR pelvis	15	Visceral abdopelvis	Whole pelvis nos	Male pelvis, female pelvis	Prostate dedicated	
MR prostate	25	Visceral abdopelvis	Prostate with/out endorectal coil		Pelvis	Without coil set up time
MR rectum	25	Visceral abdopelvis	Rectum and anal canal dedicated	Anal canal for fistula		
MR angiography abdomen	20	Visceral abdopelvis	Renal arteries	Visceral arteries, thoracoabdominal aorta		
MR angiography thorax	15	Visceral thorax	Thoracic aorta	Pulmonary arteries and veins		
MR spectroscopy (any area)	20					
MR NOS, any protocol, one structure	15					

8

Figure A1: RANZCR 2016 MR descriptors and SATs.