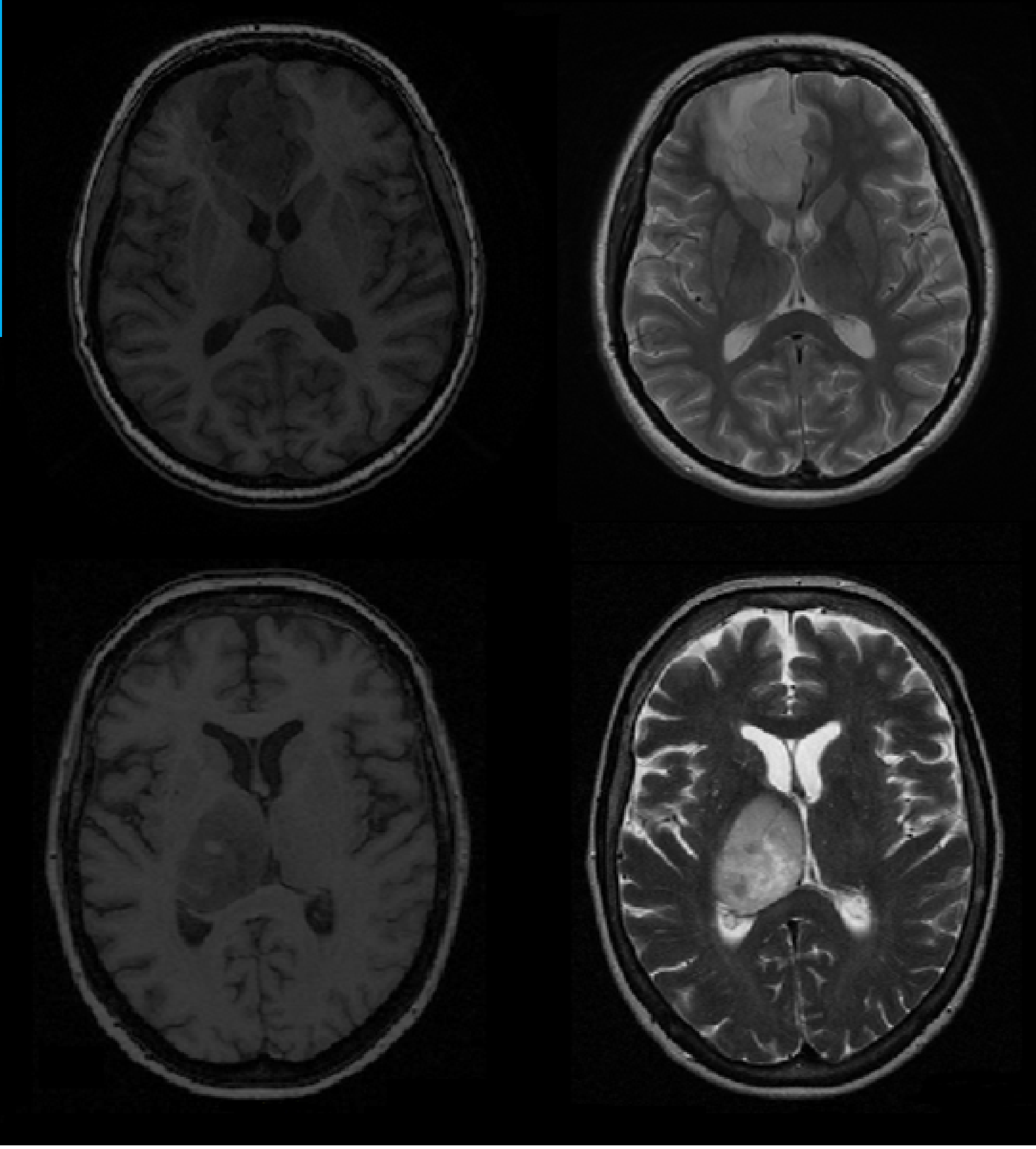
Predicting 1p/19q co-deletion status in low grade gliomas

The effect of using local binary convolutional networks

María García Sanz

Technische Universiteit Delft



TU Delft Delft University of Technology

Erasmus MC University Medical Center Rotterdam

BIGR Biomedical Imaging Group Rotterdam

Challenge the future

Predicting 1p/19q co-deletion status in low grade gliomas

The effect of using local binary convolutional networks

by

María García Sanz

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday December 18th, 2018 at 10:00 AM.

Student number:	4628632	
Project duration:	May 1, 2018 – December 18, 2018	
Thesis committee:	Prof. dr. ir. W. Niessen,	TU Delft and Erasmus MC, supervisor
	MSc. S. R. van der Voort,	Erasmus MC, supervisor
	Dr. ir. S. Klein,	Erasmus MC, supervisor
	Prof. dr. M. Loog,	TU Delft
	Dr. K. W. A. van Dongen	TU Delft
	Dr. ir. M. Staring	TU Delft and Leiden University MC

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Cover image: T1-weighted (left) and T2-weighted (right) MRI scans from a 1p/19q co-deleted low grade glioma (top) and a 1p/19q intact low grade glioma (bottom)

TUDelft

Abstract

Patients with 1p/19q co-deleted low grade glioma (LGGs) have better prognosis and react better to certain treatments than patients with intact 1p/19q LGG. Currently, information about the 1p/19q co-deletion status is obtained by means of an invasive procedure called biopsy. As an alternative, non-invasive techniques to extract this information from medical images are being studied. Recent research suggests that local binary patterns (LBPs), a textural image descriptor, are an important feature which can predict the 1p/19q co-deletion from MRI scans. In this project we report the effect of including LBP information in a convolutional neural network (CNN) to predict the 1p/19q co-deletion status in patients suffering from a presumed LGG using pre-operative MRI scans.

A combination of convolutional filters was designed and included in the CNN, resulting into local binary convolutional neural networks (LBCNNs). Three LBP descriptors, each of them representing a different textural scale, were studied, as well as the combination of the three. A default CNN without LBPs was also studied. To validate the designed filters and to study more sophisticated LBPs images like the uniform LBPs, pre-computed LBP images were directly input to the CNN. An in-house multi-institution MRI dataset consisting of 284 patients who had undergone a biopsy or resection before the treatment, and with available pre-operative T1-weighted post contrast and T2-weighted scans was used to train the different network architectures. An independent dataset consisting of 129 patients was used to validate the results. The performance of the LBCNNs was compared to the performance of the CNN.

The performance of the CNN and LBCNNs was similar, reporting an area under the receiver operating characteristic curve (AUC) ranging from 0.816 to 0.872 for the different architectures. These findings suggest that the CNN can extract information relative to LBPs by itself. In addition, pre-computed uniform LBPs report similar metrics (AUC: 0.819), suggesting that they do not add new information.

Contents

List of Figures vii

List of Tables ix

List of Abbreviations xi

1 Introduction 1
- 1.1 Clinical motivation: low grade gliomas . . . 1
- 1.2 Molecular classification of gliomas . . . 2
- 1.3 Imaging to predict the genetic status . . . 3
- 1.4 Related work . . . 4
- 1.5 Research goal and study design . . . 5
- 1.6 Thesis structure . . . 5

2 Technical Background 7
- 2.1 The radiogenomics pipeline . . . 7
  - 2.1.1 MRI data acquisition . . . 7
  - 2.1.2 Segmentation of the tumor . . . 7
  - 2.1.3 MRI data preprocessing . . . 8
  - 2.1.4 Classification and statistical analysis . . . 8
- 2.2 Convolutional neural networks . . . 9
- 2.3 Local binary patterns . . . 11
  - 2.3.1 Local Binary Convolutional Module . . . 12

3 Methods 15
- 3.1 Datasets . . . 15
  - 3.1.1 EMC/HMC dataset . . . 15
  - 3.1.2 TCIA dataset . . . 16
- 3.2 Image Preprocessing . . . 16
  - 3.2.1 Resampling and padding . . . 16
  - 3.2.2 Registration . . . 17
  - 3.2.3 Normalization . . . 17
  - 3.2.4 Selection of the MR slices and patch extraction . . . 17
- 3.3 Classification algorithm . . . 18
  - 3.3.1 Network architecture . . . 18
  - 3.3.2 Implementation of the LBP convolutional module . . . 19
  - 3.3.3 Experiments overview . . . 22

4 Results 25
- 4.1 Performance metrics of the classifiers . . . 25
  - 4.1.1 CNN classifier . . . 25
  - 4.1.2 LBCNN1 classifier . . . 26
  - 4.1.3 LBCNN3 classifier . . . 26
  - 4.1.4 LBCNN5 classifier . . . 27
  - 4.1.5 LBCNN$_{all}$ classifier . . . 28
- 4.2 Comparison between classifiers . . . 28
  - 4.2.1 Training set . . . 28
  - 4.2.2 Independent test set . . . 29
- 4.3 Comparison of the SVM with the CNN . . . 29
- 4.4 Effect of using pre-computed LBPs . . . 29
  - 4.4.1 Validation of the LBP module . . . 29
  - 4.4.2 Using uniform LBPs . . . 30

5 Conclusion and Discussion 31
5.1 Analysis of the results 31
5.2 Clinical interpretation 32
5.3 Limitations and future work of the study 32
A Acquisition parameters 35
B Construction criteria for higher order LBPs 37
C Construction of the training and validation datasets 39
D Assessing stopping criterion 41
E Training with all the data 45
F Stochascity of the CNN 47
Bibliography 49

List of Figures

1.1 T2-weighted image of a low grade glioma 1
1.2 Survival curves of grade II and III gliomas grouped according to the grade and to the genetics status 2
1.3 Illustration of the surgical procedure during a biopsy 3
1.4 T2-weighted images of 1p/19q co-deleted and not-co-deleted gliomas 4
1.5 General radiogenomics pipeline 4

2.1 In-depth radiogenomics pipeline 7
2.2 MRI modalities 8
2.3 Illustration of a CNN 9
2.4 Diagram of the construction of an LBP 11
2.5 Rotational invariant LBPs 12
2.6 Diagram of the construction of an LBP image using convolutional operations 12

3.1 Tumor patches before and after the resampling operation 16
3.2 Brain mask correction 17
3.3 Slice selection diagram 17
3.4 Diagram of the CNN architecture 18
3.5 Diagram of the convolutional module of the CNN. 19
3.6 Local Binary Convolutional Neural Network 20
3.7 LBP convolutional module 20
3.8 LBP maps in the eight encoding directions 21
3.9 Output of the layers forming the LBP convolutional module 21
3.10 Patterns to select the pixels to create LBP images of radius 3 and 5 22
3.11 Comparison of the LBP images of radius 1, 3 and 5 22

D.1 F1-score progress of the validation and test dataset 41
D.2 AUC progress of the validation and test dataset 42
D.3 Precision progress of the validation and test dataset 42
D.4 Specificity progress of the validation and test dataset 43
D.5 Sensitivity progress of the validation and test dataset 43
D.6 Accuracy progress of the validation and test dataset 44

List of Tables

3.1 Number of patients and distribution of the 1p/19q co-deletion status per subset. 16

4.1 Performance metrics of the CNN classifier on both EMC/HMC and TCIA datasets 25
4.2 Performance metrics of the LBCNN1 classifier on both EMC/HMC and TCIA datasets 26
4.3 Performance metrics of the LBCNN3 classifier on both EMC/HMC and TCIA datasets 27
4.4 Performance metrics of the LBCNN5 classifier on both EMC/HMC and TCIA datasets 27
4.5 Performance metrics of the LBCNN$_{all}$ classifier on both EMC/HMC and TCIA datasets 28
4.6 Performance metrics of the five studied architectures on the EMC/HMC dataset using T2-weighted and T1-weighted images. 29
4.7 Performance metrics of the five studied architectures on the TCIA dataset using T2-weighted and T1-weighted images 29
4.8 Performance metrics of the SVM and CNN approaches on the EMC/HMC dataset 30
4.9 Performance metrics of the CNN classifier with pre-computed LBP5 images and the LBCNN5 classifier on the EMC/HMC and TCIA datasets 30
4.10 Performance metrics of the CNN classifier with pre-computed rotational invariant and uniform radius 5 LBP images on the EMC/HMC and TCIA datasets 30

C.1 Detailed description of the distribution of classes per patient and slices in each of the folds. . . . 39

E.1 Comparison of the ensemble classifier and the classifier trained with all the data using the CNN architecture on the TCIA dataset. 45

F.1 Performance metrics of a general study of the stochascity of the CNN 47

List of Abbreviations

- AUC: Area Under the Curve
- BET: Brain Extraction Tool
- CI: Confidence Interval
- CNN: Convolutional Neural Network
- EMC: Erasmus Medical Center
- FISH: Fluorescence In Situ Hybridization
- FLAIR: FLuid-Attenuated Inversion Recovery
- FN: False Negative
- FP: False Positive
- HGG: High Grade Glioma
- HMC: Haaglande Medical Center
- IDH: Isocitrate DeHydrogenase
- LBCNN: Local Binary Convolutional Neural Network
- LBP: Local Binary Pattern
- LBP^{ri}: Rotational Invariant Local Binary Pattern
- LGG: Low Grade Glioma
- MR: Magnetic Resonance
- MRI: Magnetic Resonance Imaging
- NGS: Next-Generation Sequencing
- SVM: Support Vector Machine
- ReLU: Rectified Linear Unit
- ROC: Receiver Operating Characteristic
- TCIA: The Cancer Imaging Archive
- TCGA: The Cancer Genome Atlas
- TN: True Negative
- TP: True Positive
- VASARI: Visual AcceSAble Rembrandt Images
- WHO: World Health Organization

1

Introduction

1.1. Clinical motivation: low grade gliomas

One of the most common primary brain tumors in adults is glioma. Primary brain tumors start in the brain, unlike secondary brain tumors (also known as metastases) which are the result of cancer cells that have spread to the brain from somewhere else in the body [1]. Despite having a lower incidence rate compared to other types of cancer, brain tumors have a higher mortality rate (i.e. in 2015 in the UK, the ratio between the number of deaths caused by a breast tumor over the newly diagnosed cases was around 20%, while for brain tumors it was almost 50% [1]). Gliomas arise from two different types of glial cells (a specific type of cell in the brain) called astrocytes and oligodendrocytes [2], giving rise to a variety of types of gliomas, some being more aggressive than others.

Efforts to provide clinicians with guidelines to diagnose and treat the different types of gliomas led to the creation of a grading system by the World Health Organization (WHO) [3], grade I being the least aggressive type of glioma and grade IV the most aggressive one. The aggressiveness of the tumor was determined after the study of the tumorous tissue under a microscope, called an histopathological exam. Grade I gliomas are commonly benign [3], and thus they are not considered in this study. Grade II gliomas are called low grade gliomas (LGG) while grade III and grade IV gliomas are called high grade gliomas (HGG).

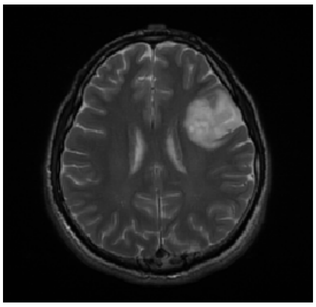


Figure 1.1: T2-weighted image of a patient with a low grade glioma (grade II glioma). The tumor is highlighted by the abnormal bright signal in the right hemisphere.

LGG, an example being depicted in Figure 1.1, have a relatively good prognosis and longer survival rate compared to HGG (while median survival for LGG can reach 14 years, HGG median survival hardly exceeds the year [4]). Patients with LGG suffer from several neurological symptoms, ranging from mild symptoms such as headaches, nausea and vomiting, to more severe problems such as seizures, changes of personality and cognitive disorders [5]. However, some patients might remain asymptomatic, depending on the size and location of the tumor.

LGG can evolve into HGG and thus clinicians need to decide which type of treatment a patient needs to receive. Selecting the appropriate type of treatment for a patient suffering from a LGG is a complicated decision.

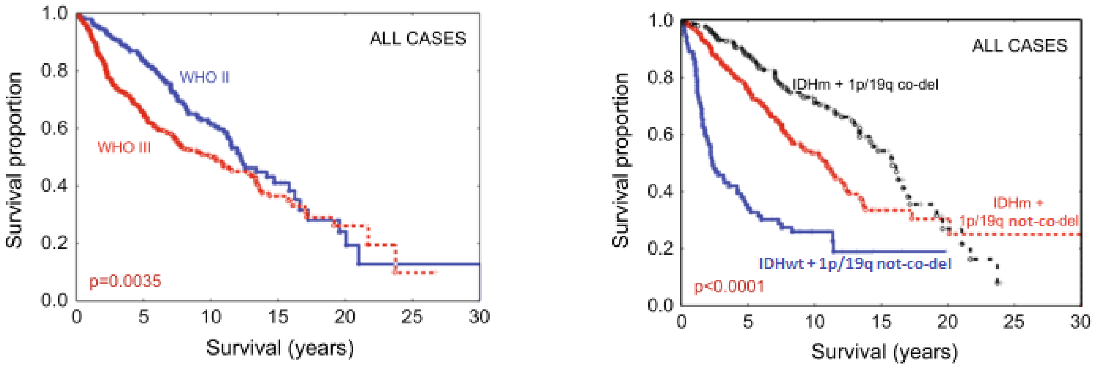
The choice is based on multiple factors, such as the location of the tumor, its tissue characteristics (i.e. histology) and the patient characteristics (i.e age, symptoms) [5]. Maximal safe resection, the process by which the neurosurgeon removes the maximal accessible tumorous tissue without damaging the neurological status of the patient, is usually the preferred first step adopted by clinicians [6, 7]. Radiotherapy and chemotherapy treatment follow the resection in some cases (especially if the tumor appears to grow after the resection), to completely remove the tumorous tissue from the inaccessible regions during the surgery [5]. However, sometimes clinicians prefer to wait and watch the evolution of the tumor before proceeding with the treatment, due to the risks that it entails [5]. Surgery and radiotherapy treatments may damage the healthy tissue in the brain, affecting the neurological status of the patient and even inducing new brain tumors. Therefore, having a good stratification criteria to classify LGG would help clinicians to improve the risk-assessment of the patients.

Histopathological characterization of a tumor, as previously mentioned, mainly relies on the observance of microscopic features through light microscopy [8], providing a degree of subjectivity in the interpretation of the tissue sample. This can lead to inter-observer variability in the determination of the grade [9–11], directly interfering in the process of selecting the proper treatment for a patient. Recent research in the field of genomics suggests that molecular classification of gliomas (based on the study of the genes of the tumors) provides better stratification than classical histopathological classification (based on the study of the tissue and cells of the tumors). Therefore, in order to help clinicians with their decision making step, it would be beneficial to know the molecular classification of a glioma before deciding on the type of treatment.

1.2. Molecular classification of gliomas

Recent advances in genetic research are shedding a new light on the molecular structure of gliomas, contributing to the characterization of these tumors. As mentioned above, the assessment of the aggressiveness of a glioma was historically done through histopathological exams. But since 2016, two molecular biomarkers have been included in the WHO guidelines [8]: the citric-acid-enzyme isocitrate dehydrogenase (IDH) mutation status and the 1p/19q co-deletion status. The term 1p/19q co-deletion accounts for the simultaneous deletion of the short arm (i.e. p) of chromosome 1 and the long arm (i.e. q) of chromosome 19. This new molecular classification of gliomas is categorized in three groups: (1) IDH-wild type (the most aggressive one, with survival characteristics similar to HGG), (2) IDH-mutant and 1p/19q not-co-deleted, (3) IDH-mutant and 1p/19q co-deleted (the one with better prognosis and survival characteristics similar to LGG, which also has shown a better response to radiotherapy combined with chemotherapy treatment [12, 13]).

In one study encompassing 558 grade II and grade III gliomas, Olar et al. compared the overall survival of the patients based both on the histology (WHO grades) and on the molecular analysis (IDH status and 1p/19q co-deletion status), the results being shown in Figure 1.2. While in Subfigure 1.2a, the curves corresponding to the WHO grade classification are very similar, the curves in Subfigure 1.2b corresponding to the three categories of the molecular classification have different behaviours. The study proves the power of the molecular classification over the histopathological classification for patient stratification.



(a) Histopathological classification.

(b) Molecular classification.

Figure 1.2: Overall survival of 558 grade II/III gliomas stratified by (a) the WHO grade (b) the IDH mutation status (*IDHm*: IDH mutation; *IDHwt*: IDH wild type) and the 1p/19q co-deletion status (*not-co-del*: not-co-deleted; *co-del*: co-deleted) [14]. Molecular biomarkers show better patient stratification than WHO grades.

However, molecular exams are limited by their need for tissue samples from the tumor, extracted by means of an invasive technique called biopsy, illustrated in Figure 1.3. Performing a brain biopsy is a risky procedure, since part of the skull needs to be removed and the brain must be handled delicately. In addition, obtaining sufficient tumorous tissue containing sufficient DNA to perform a molecular exam is not always trivial. In fact, The Cancer Genome Atlas (TCGA) reported in their first study about gliomas that only 35% of the available samples contained sufficient tumorous tissue with adequate DNA material to perform conclusive molecular exams [15]. Moreover, some studies even suggest that tumors might be genetically heterogeneous [16–19], and therefore biopsies may fail in assessing the complete anatomic and physiologic profile of the tumor. In addition, monitoring the molecular profile during treatment through tissue analysis and multiple biopsies is not feasible due to the costs [20] and the risks for the patients. Consequently, performing molecular exams to assess gliomas is still a challenging procedure to include in the daily clinical practice.

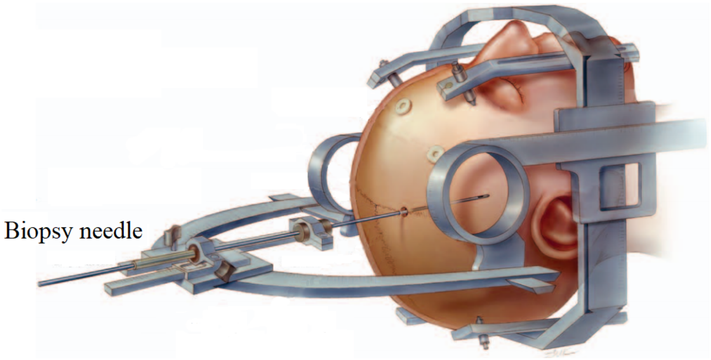


Figure 1.3: Representation of the surgical procedure to perform a biopsy (illustration modified from [21]). A hole is drilled through the skull of the patient to insert the biopsy needle. A rigid frame around the head helps guiding the needle into the brain.

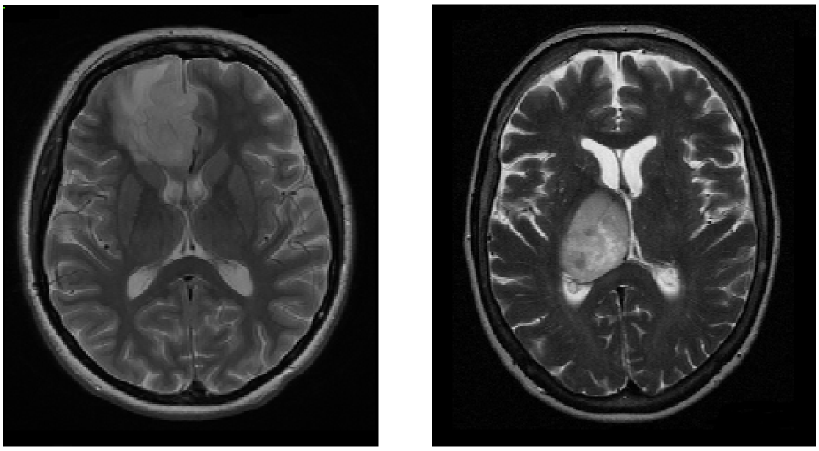
1.3. Imaging to predict the genetic status

Medical imaging techniques are widely used by clinicians to detect the presence of a tumor and to monitor the patient throughout the treatment. Human tissues often show a variety of distinctive attributes on radiographic images, depending on the modality employed. For the particular case of brain tumors, Magnetic Resonance Imaging (MRI) provides sufficient contrast to distinguish between the different soft tissues in the brain, and even between the different textures of the tumors (such as necrotic areas and edema areas) [22]. The different modalities of MRI provide clinicians with a variety of images from which to extract an extensive and complementary description of the tumor.

In fact, a lot of effort is put in the development of lexicons derived from medical images. Lexicons are dictionaries containing imaging features whose aim is to provide clinicians with a set of guidelines to assess the aggressiveness of a tumor from images. For the case of brain tumors, radiologists benefit from the Visually AcceSAble Rembrandt Images (VASARI) lexicon [23], containing more than 20 semantic features related, for example, to the location of the tumor, and the presence of hemorrhage among others. For the specific case of the 1p/19q co-deleted glioma, the following imaging characteristics have been related to its presence (as seen in Subfigure 1.4a): having tumors in the frontal lobe of the brain, having tumors with indistinct borders, having tumors with an heterogeneous signal intensity on T2-weighted images, and having tumors with cortical and subcortical infiltration [4].

However, despite the effort put in the standardization of the descriptive process, lexicons and semantic features are hampered by the inter-observer variability, since a specific feature can be graded differently by two different observers. In addition, the information the radiologist extracts from the image is limited, as their assessment is restricted to the human eye and the analysis is merely qualitative.

Radiomics is an emerging field which intends to counteract the inter-observer variability by giving an objective, quantified and repeatable description of tumors with the help of computational algorithms. First, the algorithms extract a large amount of quantitative imaging features such as shape, texture and intensity



(a) 1p/19q co-deleted glioma. **(b)** 1p/19q not-co-deleted glioma.

Figure 1.4: Comparison of T2-weighted images of 1p/19q co-deleted (a) and not-co-deleted gliomas (b). The 1p/19q co-deleted glioma is located in the frontal lobe and has indistinct borders with infiltrations. The not-co-deleted glioma is located in the limbic lobe and has definite borders.

histogram-based statistics from the images of the tumors [24–28]. Then, to reduce redundancy, the most important features for the studied task are selected. Ultimately, radiomics methods employ statistics and machine learning algorithms to create models that try to predict a certain clinical outcome such as the type of tumor and its aggressiveness. A subfield of radiomics, radiogenomics, links the previously mentioned quantitative imaging features with genomic signatures to create models that predict the genetic status of a tumor [29, 30]. Therefore, the radiogenomics pipeline, depicted in Figure 1.5, is a potential tool for the prediction 1p/19q co-deletion status without the need of performing a biopsy.

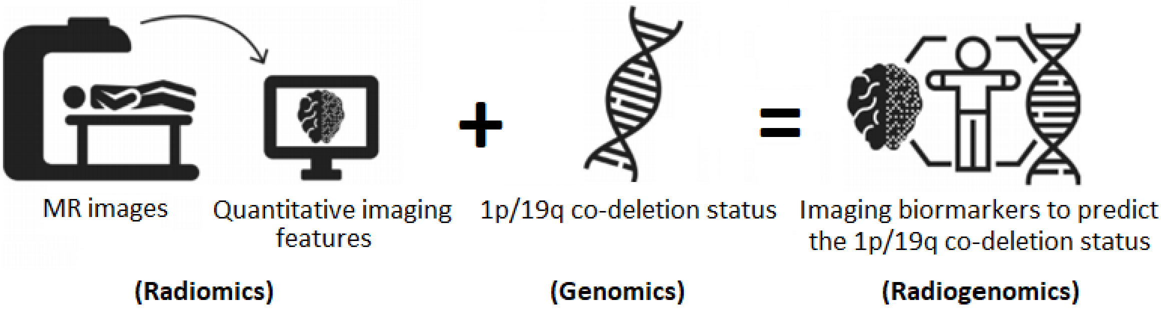

MR images
Quantitative imaging features
+
1p/19q co-deletion status
=
Imaging biormarkers to predict the 1p/19q co-deletion status
(Radiomics)
(Genomics)
(Radiogenomics)

Figure 1.5: General radiogenomics pipeline: medical images such as MR images are employed to extract quantitative imaging features by means of computational algorithms (Radiomics), which combined with genetic labels such as the 1p/19q co-deletion status (Genomics) allow to obtain imaging biomarkers to create models which predict the 1p/19q co-deletion status (Radiogenomics). Illustration based on Jansen et al. [31]

1.4. Related work

Some studies have already proven the power of the radiogenomic approach to predict the 1p/19q co-deletion status using classic machine learning algorithms [32–35]. For example, van der Voort et al. [32] trained a support vector machine (SVM) using an in-house database of 63 patients suffering from a LGG. Shofty et al. [33] trained and compared the performance of 17 different algorithms employing a dataset of 47 patients with LGGs. To our knowledge, only one publicly available study validated their algorithm on an independent test set, but using only 5 patients [35].

In the past years, the interest in deep learning approaches has increased. The main characteristic of a deep learning algorithm is its ability to automatically extract and select the relevant features (feature extraction and selection being two important steps in the radiomics pipeline, as mentioned above). A particular type of deep learning algorithm, the convolutional neural network (CNN), has been proven to outperform classical machine learning algorithms in classification problems involving images, notably in the computer vision field [36]. Following that trend, some studies have explored CNN algorithms to predict the 1p/19q co-deletion status [37, 38]. Akkus et al. [37] employed a single center LGG dataset of 159 patients to train a multi-scalar CNN. Chang et al. [38] trained a residual CNN using 119 patients suffering from LGGs. Still, none of them provided a validation over an independent dataset.

Nevertheless, CNN have thousands of trainable parameters and thus large amounts of data to prevent overfitting (i.e. to prevent the network from memorizing the dataset instead of learning) are required. To our knowledge, no sufficiently large and assorted glioma dataset is available yet. In addition, CNN suffer from the intensity variability among the MRI scans. CNNs learn the features of an image based on the intensity relationship of its pixels. But the gray values computed by the MR scanners are not absolute (i.e. no direct measurements), they are weighted, resulting in having a different intensity scale per MR image. Therefore, solutions for the intensity variability problem need to be proposed when working with CNNs.

1.5. Research goal and study design

Our research project presents a CNN approach to predict the 1p/19q co-deletion status from non-invasive MR images. We propose to use knowledge from the SVM studies in the design of the CNN to reduce the effect of the intensity variability issue when predicting with MRI images. In addition, we have studied the robustness of our classifier by validating the results on an independent dataset.

The project studied the effect of incorporating a combination of convolutional filters in the CNN to guide it to learn features based on local binary patterns (LBPs), a textural descriptor that has been proven to contribute to the predicton of the 1p/19q co-deletion status from MRI scans (this information was extracted from an unpublished study using an SVM as the classifier, which is the continuation of the work done in van der Voort et al. [32]). An LBP descriptor can easily be implemented using convolutional filters, whose size and weight distribution determines the scale of the LBP image. What is more important, LBPs are grayscale invariant, which could overcome the intensity variability of the MRI scans. By incorporating the LBP module, we are expecting to improve the performance of the CNN and to increase its capacity to generalize on unseen data.

The design of the study is as follows. The first step consisted of creating the LBP module employing non-trainable convolutional filters to be inserted in a CNN, resulting in local binary convolutional neural networks (LBCNNs). The effect of having pre-computed LBP images directly as a second channel in the classifier, without having to use the LBP module, was then studied. The purpose of this step was to validate the LBP module, which approximates the LBP descriptor, and to study the effect of having more sophisticated LBP images, like the uniform LBP images. The second step was focused on the implementation of the preprocessing pipeline to extract patches of the tumors from the MRI images. The third step consisted of designing a suitable 2D-CNN which was able to report performance metrics similar to the SVM of the aforementioned unpublished study. The main goal of the project has been the study of the effect of different LBCNNs. Three different types of LBP descriptors, each of them with a different scale, were studied: LBP descriptor of radius 1, 3 and 5, as well as the combination of the three. The default CNN architecture was also studied. To train the different architectures, an in-house multi-institution dataset consisting of 284 patients suffering from a presumed LGG was used (a presumed LGG is a glioma which is suspected of being an LGG when looking at the medical images but which is not confirmed by the histopathological and molecular exams). Both pre-operative T1-weighted post-contrast and T2-weighted scans were available. All the experiments were first performed with the T2-weighted images as input for the classifier. Then we employed both T1-weighted and T2-weighted images, expecting that this approach provides with better performance as reported in van der Voort et al. [32]. All the derived CNN architectures were tested on an independent, single-institution, publicly available dataset from The Cancer Imaging Archive (TCIA), to validate the results obtained during the training and to assess the robustness of the classifier.

1.6. Thesis structure

The thesis is divided into five chapters. Chapter 2 introduces the reader to the radiogenomic approach and to the LBPs, as well as provides a general description of a CNN. Chapter 3 describes the methodes used in this study, including the dataset, the pre-processing pipeline, the designed CNN and the experiments that we have designed to test the hypothesis presented in the previous section. Chapter 4 shows the results of the performed experiments. The thesis concludes with Chapter 5, which presents the conclusions extracted from the results, as well as the discussion section. It also includes a section about the limitations of our study which suggests the next steps that can be taken in this field.

2

Technical Background

2.1. The radiogenomics pipeline

Radiogenomics is a novel technique which has become an important topic of research in the oncological field. The purpose of the radiogenomics approach is to obtain an exhaustive description of a tumor by extracting quantitative features from medical images and link them to a genetic mutation. Figure 2.1 shows the pipeline for a radiogenomics study which employs a CNN as a classifier.

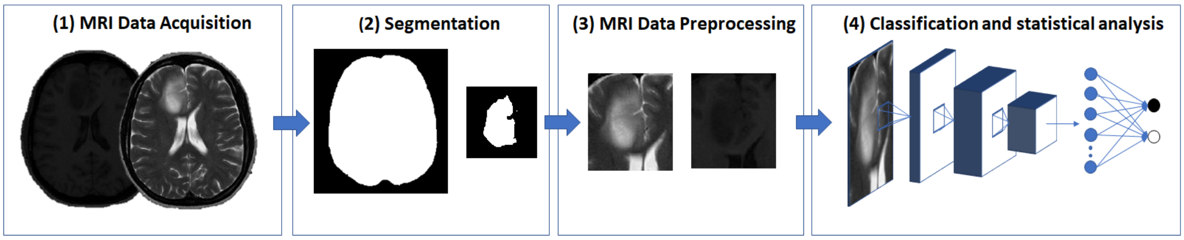

(1) MRI Data Acquisition
(2) Segmentation
(3) MRI Data Preprocessing
(4) Classification and statistical analysis

Figure 2.1: Steps performed in the radiogenomics pipeline: (1) acquisition of the different MR images of the tumor; (2) segmentation of the tumor and the brain; (3) preprocessing of the MR images: registration, normalization and extraction of the tumor patches; (4) construction of the CNN classification model.

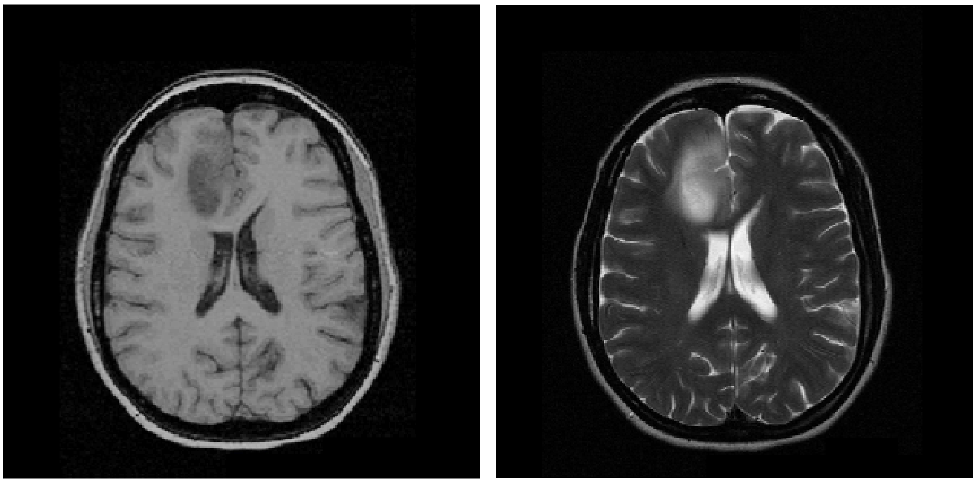
2.1.1. MRI data acquisition

The first step in the radiogenomics pipeline is to obtain images from the studied tumors and its environments. As mentioned in the Section 1.3 of the introduction, MRI is the preferred technique to study brain tumors. Typically, images from different modalities, such as T1-weighted imaging and T2-weighted imaging (see Figure 2.2) are employed, since they provide complementary information to characterize the tumor.

An MR image is created by electromagnetically exciting hydrogen atoms contained in the tissues and measuring the electromagnetic field they produce as they return to their resting state [39]. MR images thus map the relaxation properties of the hydrogen atoms of different tissues (recovery time for the T1-weighted modality in the longitudinal plane, and decay time for the T2-weighted modality in the transversal plane). Therefore, the gray values of an MR are strongly dependent on the scanner used and do not directly reflect tissue characteristics but only a weighted value. Further normalization techniques in the preprocessing step will be required to compare different MR scans.

2.1.2. Segmentation of the tumor

The next phase of the radiogenomics pipeline is to segment the tumorous tissue. If more than one image modality is employed, the segmentation mask is extracted from one of them and registered to the rest. Semi-automatic methods are the current standard in the practice [40, 41]. In this case, an expert works with a software to produce the segmentation mask of the tumor. Compared to pure manual segmentations, where



(a) T1-weighted image of an LGG. **(b)** T2-weighted image of an LGG.

Figure 2.2: MRI modalities employed in the radiomics pipeline depicting different radiographic characteristics. The contrast on the T1-weighted image has been adjusted for visualization purposes.

only the expert contributes to the process, semi-automatic approaches increase the reproducibility and speed up the segmentation process [42]. In addition, semi-automatic methods work better for tumors with fuzzy or spiculated edges than fully automatic methods (where only a computer contributes to the process) [40].

Brain image segmentations are also required to be able to remove the skull of the MR scans. This is an important step for the following normalization step, as we are only interested in quantifying the relationships between the gray values of the brain tissues. Fully-automatic methods are the standard procedures, since there are many online libraries specialized in this type of tasks, the FSL library being the most employed.

2.1.3. MRI data preprocessing

CNNs create the features for the classification step by linearly combining the gray values of the input images. To be able to construct meaningful features, the images need to be aligned. This is achieved by means of registration algorithms, which employ algebraic transformations to ensure that both images are in the same space.

The next step in the preprocessing phase is to normalize the gray value intensity of the MR scans. The first task, as previously mentioned, is to remove the skull using the brain mask. Since the CNN algorithm is going to be a 2D network, normalization is performed per slice and not per the whole MR scan. There is no standard procedure to normalize an MR image, but one of the most employed is the Z-score. This algorithm consists of subtracting from the gray value g_i of the MR slice the mean value μ of the slice and divide it by the standard deviation σ of the slice:

$$z_i = \frac{g_i - \mu}{\sigma} \tag{2.1}$$

Finally, to help the CNN with localizing the tumor in the image, only slices containing the tumor are fed into the algorithm. To further ease the task, only patches of the tumor, and thus not complete brain images are input to the CNN.

2.1.4. Classification and statistical analysis

In the classification step, the preprocessed MR slices are input to a CNN (further described in the next section), which employs labeled data to train the algorithm. Performance metrics are used to evaluate the performance of a classifier [43]. They provide with parameters to compare different types of classifiers. The following metrics allow the user to understand the context of the classification by combining the values extracted from the confusion matrix (which are the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN)):

- Accuracy:
$$\frac{TP+TN}{TP+FP+TN+FN}$$
- Sensitivity or true positive rate or recall:
$$\frac{TP}{TP+FN}$$
- Specificity:
$$\frac{TN}{TN+FP}$$
- Precision:
$$\frac{TP}{TP+FP}$$

However, datasets can suffer from class imbalance and parameters such as the accuracy are very sensitive to these sort of problems. To cope with this problem, it is wise to report parameters that work better in front of the class imbalance. Examples of such parameters are:

- Area under the curve (AUC): The Receiver Operating Characteristic (ROC) curve plots the false positive rate (FP/(FP+TN)) against the true positive rate. The area under this curve is an estimation of the accuracy of the classifier, 0.75 being the lower threshold from which we can consider that an algorithm is performing well [44].
- F1-factor:
$$2\cdot\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}=\frac{2TP}{2TP+FN+FP}$$

2.2. Convolutional neural networks

The characteristic of a deep learning algorithm compared to the rest of machine learning algorithms is its ability to extract the relevant features from the images by itself. Thus, a network can be divided into two parts: a series of stacked layers in charge of extracting the key features, and the final layers in charge of making the prediction (as seen in figure 2.3).

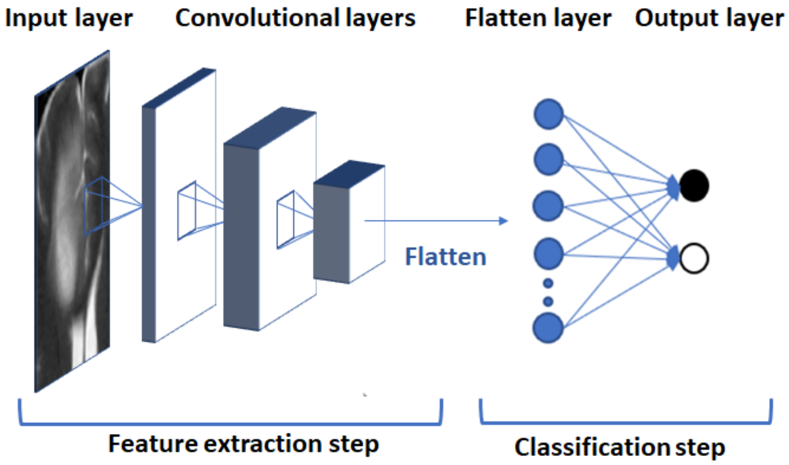


Figure 2.3: Example of CNN. Patches from the input image are linearly combined using convolutional filters to create the feature maps. The number of employed convolutional filters determines the amount of obtained feature maps (depicted with the width of each layer). This process corresponds to the feature extraction step. The produced feature maps are commonly flattened to create the feature vector which is used by the last layer to classify the input image. This is the classification step.

A layer of a neural network is made of neurons. The value of each neuron y_i is a combination of the values x_j of the neurons of the previous layer:

$$y_i = f\left(\sum_{j=1}^{n_x} \omega_{ij} x_j + \sigma_j\right) \tag{2.2}$$

ω being the weights between neurons, σ being the biases, n_x being the number of neurons of the previous layer and $f(x)$ being an activation function. The learning process consists of finding the optimal weights ω and biases σ from all the layers that minimize the loss function $L(\omega,\sigma)$ using labeled samples. The loss function models how far the predictions of the classifier are from the ground truth labels. It can thus be understood as the quantification of the error of the classifier. In binary classification, where there are only two classes, 0 and 1, one of the most commonly employed loss functions is the cross-entropy:

$$L = -y_1 ln(\bar{y}_1) - y_0 ln(\bar{y}_0) \tag{2.3}$$

$ln(x)$ being the natural logarithm, y_0 and y_1 being the ground truth labels and $\bar{y}_0$ and $\bar{y}_1$ the predicted probabilities by the network. The process by which the loss function is minimized is called the back propagation algorithm [45], based on the gradient descent algorithm [46]. When a training example is input into the network, an error value can be calculated from the loss function. The error of the last layer is back-propagated layer by layer and it is used to compute the gradients of the loss function with respect the weights and biases of all the layers. These gradients indicate the amount by which each weight and bias needs to be tuned to get closer to the minimum of the loss function. In practice, one computes the average gradient once a batch of samples has been propagated in the network instead of only a single sample, the purpose being to increase the accuracy by which the weights and biases are updated. This method is called mini-batch gradient descent [46].

Training a neural network is thus a complicated procedure, since the inputs of each layer are affected by the previous layers, requiring a fine-tuning of the hyperparameters. In addition, the inputs during training are constantly changing, since both the weights and biases of the previous layers are being tuned. The change in the input distribution of the layers is known as the covariate shift [47], and it slows down the learning process. Batch normalization layer was introduce by Ioffe et al. [48] to reduce the effect of the covariate shift. This layer normalizes the input distribution by adjusting the mean and the standard deviations of each neuron using the mini batch mean and standard deviation. To avoid introducing major changes that would affect the output prediction, two learnable parameters that shift and scale the activation value of each neuron are introduced. Batch normalization layers are also said to act as a sort of regularizer, since they introduce noise to each of the neurons. Therefore, they contribute to reduce overfitting.

CNN is a specific type of deep learning algorithm which uses convolution operations to propagate the information from one layer to the other [49]. Therefore, in this type of networks, rather than having individual and independent weights that connect each input neuron to each output neuron of two consecutive layers, the weights are shared between neurons, leading to sparse interactions and a reduced number of training parameters. Commonly, a convolutional layer can be interpreted as a filter operation with a series of kernels, where the information of the previous layer is called the input of the layer, and the output result is called the feature maps. Each convolutional kernel has three dimensions, width, height and depth, the depth corresponding to the amount of feature maps in the input layer. The number of output feature maps corresponds to the number of kernels employed. Research in the computer vision field suggests that the weights of the initial convolutional kernels learn to detect general morphological features such as edges, common to the majority of images, while deeper layers extract particular features from the employed training dataset [50].

Pooling and activation layers are commonly employed in CNNs. Pooling layers reduce the size of the feature maps while activation layers quantify the amount of information that is passed from a neuron to another (like a sort of weighted switch). In a pooling layer, the value of an output pixel is a statistical combination of their neighbors. The $max(x)$ function is typically the preferred statistical function, but there are other variants such as the $mean(x)$ function or the $min(x)$ function. Global average pooling is a particular type of pooling layer employed right before the classification layers to avoid using flatten layers, a type of layer which simply unrolls the pixels of each feature map creating a long single feature vector. Global average pooling layer computes the average value of each feature map. As an example, consider having at the end of the pipeline 64 feature maps of 4x4 pixels. While a flatten layer would create a feature vector of 1024 units, global average pooling creates a feature vector of 64 units. Therefore, the number of training parameters in the classification layers is reduced.

Layers introducing the activation function $f(x)$ complete the basis of a CNN. The most employed activation function is the rectified linear unit (ReLU), expressed by: $R(z_i) = max(0, z_i)$, z being the linear combination of the values of the previous layer weighted with the weights ω and biases σ ($z_i = \sum_{j=1}^{n_x} \omega_{ij} x_j + \sigma_j$).

2.3. Local binary patterns

LBPs are computationally efficient texture descriptors widely used in the computer vision field [51]. Developed by the end of the 90s, they are invariant to monotonic transformations of the gray levels [52, 53]. A monotonic transformation is a numerical change to a set of numbers so that the rank between these numbers is preserved (i.e. if the voxel i has an intensity greater than a voxel j, after a (positive) transformation f, $f(i)$ will still be greater than $f(j)$). The property of being invariant to a monotonic transformation implies that the output of an LBP operator will be the same as long as the rank of the intensities is kept. This property is very important in the field of medical image analysis, above all when dealing with MR images, due to the intensity variability issue. Therefore, one could compare the LPBs of two MR images (of the same modality, i.e T1-weighted post contrast) from the same disease but performed with MRI machines from different manufacturers and with different image protocols.

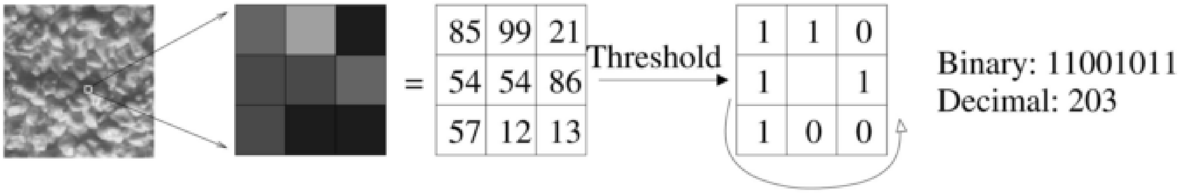


Figure 2.4: The LBP operator. First the image is divided in patches of 3x3. Then, the difference between the central pixel and the neighbors of each patch is computed (not shown in the image), and assigned to 0 if the result is negative and to 1 otherwise (shown in the image as the "Threshold" label). Finally, an encoding direction is predefined (marked with the arrow) to construct the binary label, which can be translate to a decimal number [51].

The technique to construct an LBP label is as it follows. Commonly, the image is divided in patches of size 3x3 and the difference between the central pixel and the neighbors of each patch is computed. A threshold function $s(x)$ is then applied:

$$s(x) = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{if} \quad x < 0 \end{cases} \tag{2.4}$$

which changes to 1 the positive or null results and to 0 the negative ones. An encoding direction needs to be predefined, meaning to select the pixel from which you start reading the binary number. Once decided, the binary number can be constructed, providing the patch with a label (see Figure 2.4). The histogram of all the labels (previously transformed into decimal values) is then used as a texture feature.

The LBP operator can be used to extract textures in other scales. In this case, one can define the LBP by setting the value of two parameters, P and R, corresponding to the number of points forming a circle and the radius of the circle respectively (see figure 2.5). Following the aforementioned dynamics, the label is constructed by subtracting the intensity value g_0 of the pixel falling into the center of the circle and the intensity value g_i of the other points forming the circle. When a point appears to fall outside the center of a pixel, its gray value is computed by using interpolation. Equation 2.5 depicts the final formula of the LBP descriptor.

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_0) 2^i \tag{2.5}$$

The LBP operator can output 2^P different LBP maps, depending on the starting point of the encoding direction. If the image is rotated, the intensity values g_p will also move, changing the LBP for a specific encoding direction. To remove the effect of rotation, and thus to reduce the number of possible LBP patterns to only one, a rotational invariant version of LBP (LBP^{ri}) is proposed [53]:

$$LBP^{ri}_{P,R} = min\{ROR(LBP_{P,R}, i) \quad \| \quad i = 0, 1 \ldots P-1\} \tag{2.6}$$

where ROR(x,i) is the function that changes the encoding direction by shifting the most significant bit x of the LBP to the right i times. Figure 2.5 shows 16 of the 36 $LBP^{ri}_{P,R}$ with $P = 8$. The pictures suggest why LBPs can be seen as feature detectors, as they can be interpreted as edge detectors (i.e. LBP number 4), or bright spot detectors (i.e. LBP number 0).

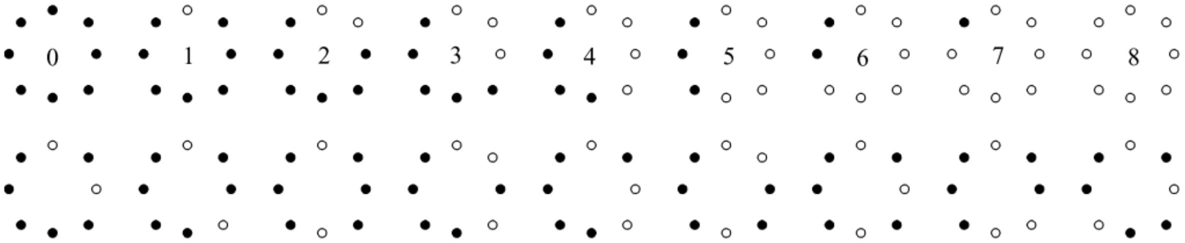


Figure 2.5: 16 of the 36 rotation invariant binary patterns of the circular set of LBP with $P = 8$. The first row corresponds to the uniform patterns [53].

Ojala et al. in [53] noticed that almost 90% of the LBPs in texture images where uniform. A LBP is said to be uniform when there are two or less bitwise transitions (from 1 to 0 and vice versa) in the binary label (see top row of Figure 2.5). Therefore, one can further reduce the the family of $LBP^{ri}_{P,R}$ descriptors to uniform $LBP^{ri}_{P,R}$ to describe the texture of an image.

2.3.1. Local Binary Convolutional Module

In their study, Juefei et al. [54] proved how to express an LBP descriptor with convolutional operations to be inserted in a convolutional neural network. Let us consider the simplest case of LBPs with $P = 8$ and $R = 1$. The initial difference operation between the center pixel and the neighbors can be seen as a set of convolutions with 8 sparse 3x3x1 kernels $\mathbf{b}_i$ (1 being the depth of the convolutional kernel), where the center pixel is -1 and one of the neighbors is 1 (the position of this positive value is switched in each of the kernels, as seen in the left part of Figure 2.6).

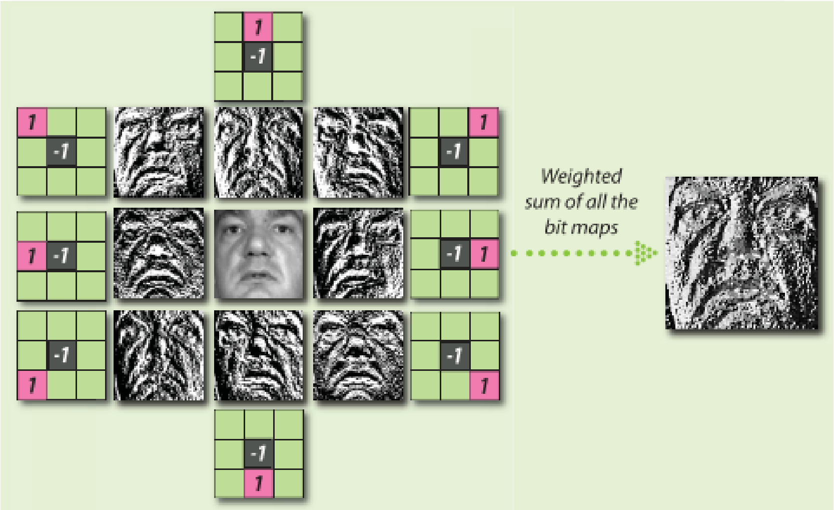


Figure 2.6: LBP operation expressed with convolution operations [54]. The input image is convolved with the set of eight 3x3 sparse kernels, containing -1 in the center pixel (in dark green) and 1 in one of the remaining neighbors (in pink) (the rest of values being 0, in light green). The results of the convolutions are also shown in the image right next to each kernel. After the non-linear operation with the Heaviside step function, the 8 bit-maps are convoluted with the binary weights $(2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7)$ so as to obtain the weighted sum which produces the LBP image. The order of the binary weights sets the encoding direction.

The threshold operation can be performed with a non-linearity $f(x)$, more specifically the Heaviside step function. Finally, the encoding operation can be expressed as a 1x1x8 convolution with the set of binary weights $\mathbf{v}_i$ $(2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7)$, resulting in the LBP image shown in the right side of Figure 2.6. The order of the binary weights sets the encoding direction (another encoding direction is for example: $2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^0$). The reformulation of the LBP operator with convolutional filters is:

$$LBP = \sum_{i=0}^{7} f(\mathbf{b}_i * \mathbf{x})\mathbf{v}_i \tag{2.7}$$

3

Methods

3.1. Datasets

3.1.1. EMC/HMC dataset

The EMC/HMC dataset comprises a total of 284 patients suffering from a low grade glioma. To be included in the study, patients were required to be at least 18 years old and have had a biopsy or resection between October 2002 and March 2017. In addition, the 1p/19q co-deletion status had to be known and pre-operative T2-weighted and T1-weighted post-contrast scans had to be available.

Patients from this dataset have been treated at two different hospitals, the Erasmus Medical Center of Rotterdam (EMC) and the Haaglande Medical Center of The Hague (HMC). However, some of the EMC patients have been diagnosed in other centers, meaning that the pre-operative MRI scans were acquired elsewhere (from a list of fifteen different clinics).

Fluorescence In Situ Hybridization (FISH) [55] and Next-Generation Sequencing (NGS) techniques [56] were used by molecular biologists to prove the mutation status.

Image acquisition and segmentation

The MRI machines employed to scan the patients were from different vendors, namely General Electric, Philips and Siemens. The ranges from the acquisition parameters (voxel spacing, matrix size, echo time, repetition time, number of slices, slice thickness and field strengths) are stated in Appendix A.

An expert neurologist with 10 years of experience visually inspected all the scans to ensure that the included ones had sufficient axial resolution and no artifacts. Only presumed LGG were included in the study. A tumor was considered presumed LGG if no or mild enhancement appeared in the pre-operative T1-weighted post-contrast scan.

A semi-automatic approach was employed as the segmentation protocol. The tumors were segmented by two different people using the the `ITKSnap` standard toolbox. Segmentation was done on the Fluid-attenuated inversion recovery (FLAIR) modality if available (119 patients), otherwise directly on the T2-weighted modality (165 patients). In the former case, the FLAIR scan was first registered to the T2-weighted scan space by means of a two-step transformation: a first rigid transformation and a second affine transformation. The metric employed in both cases was the advanced mattes mutual information [57]. The registration procedure was implemented by a technical expert using the `SimpleElastix` library [58]. Later registration of the T1-weighted scans on the T2-weighted scans (check Section 3.2.2), removes the need of obtaining T1-weighted tumor masks. All the tumor masks were inspected by the neuroradiologist expert.

Brain masks were also created and checked by the same technical expert for both T1-weighted and T2-weighted modalities. The Brain Extraction Tool (BET) of the FSL library with a setting of 0.5 was employed for this task.

3.1.2. TCIA dataset

An additional single center patient cohort extracted from the TCIA "LGG-1p19qDeletion" [59] is employed as testing set. This dataset contains co-registered T2-weighted and T1-weighted preoperative scans from 159 patients suffering from a histopathologically proven LGG, using the FISH technique. From the 159 available patients, only 129 were included in our dataset based on the previously mentioned criteria of selection. The process was again supervised by the same expert neuroradiologist. Table 3.1 summarizes the number of patients and the 1p/19q co-deletion distribution among the two datasets.

The full segmentation of the tumors was done by the expert neuroradiologist using `ITKSnap` on the T2-weighted scan. The same mask can be employed on the T1-weighted image since both scans are already co-registered. Brain masks were individually extracted for each modality by the same technical expert using the same aforementioned approach and the FSL library.

Table 3.1: Number of patients and distribution of the 1p/19q co-deletion status per subset.

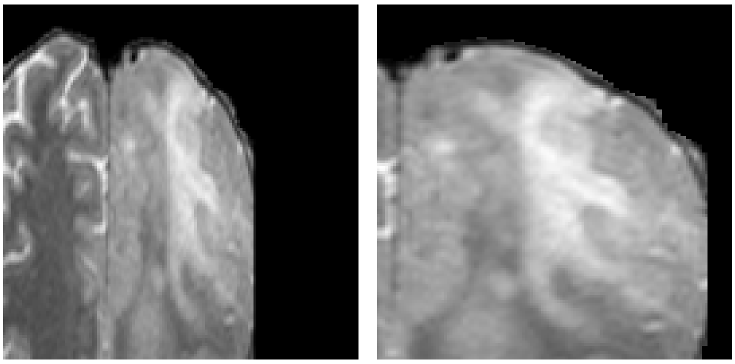
	EMC/HMC	**TCIA**
Number of patients	284	129
1p/19q not-co-deleted tumors	184	44
1p/19q co-deleted tumors	100	85

3.2. Image Preprocessing

3.2.1. Resampling and padding

The T2-weighted scans with different voxel size in the axial plane were resampled to ensure that the extracted axial slices had the same voxel size. CNNs do not distinguish between rectangular and squared pixels, since an image is perceived as a matrix of gray values. Therefore information is lost when dealing with non-squared voxels. The resampling operation was done using the `ITK` library by setting the highest resolution direction as the matching value and with a linear interpolator. The resampling operation was also applied on both tumor and brain masks. Figure 3.1 shows an example of the effect of the resampling operation when extracting the tumor patch: while Subfigure 3.1a depicts a squeezed tumor with an abnormally big patch size, Subfigure 3.1b depicts the tumor with the desired patch size.

A zero-padding strategy was implemented if the studied T2-weighted scan had a different number of voxels in the directions of the axial plane. The purpose of this step is to prevent the loss of information during the rotations implemented at the data augmentation step, explained in Section 3.3.1. Thus, we ended up working with scans containing square slices with square voxels.



(a) Tumor patch before resampling. **(b)** Tumor patch after resampling.

Figure 3.1: Effect of the resampling operation: the resampled tumor is no longer squeezed and the patch possesses the desired dimensions.

3.2.2. Registration

T1-weighted scans have been registered on the T2-weighted scans. The two step registration procedure with `SimpleElastix` mentioned at Section 3.1.1 is used to map the T1-weighted scan on the T2-weighted scan: an initial rigid registration followed by an elastic registration using a b-spline filter, both having the advanced mattes mutual information as a metric. Once having both scans in the T2-weighted space, both T2-weighted tumor and brain masks can be directly applied on the registered T1-weighted scan.

3.2.3. Normalization

Brain masks volumes were used to remove the skull from the scans. An automatic correction for the extreme cases where (part of) the tumor laid out of the brain mask was included into the pipeline: in case that more than 20% of the area of the tumor is excluded from the brain mask, a new brain mask is derived by adding the old mask and the tumor mask (see Figure 3.2). A dilation filter together with a fill-in-the-holes filter from the `SciPy` library was employed to smoothen the new brain mask. The masked scans were then normalized using the Z-score algorithm of the `ITK` library.

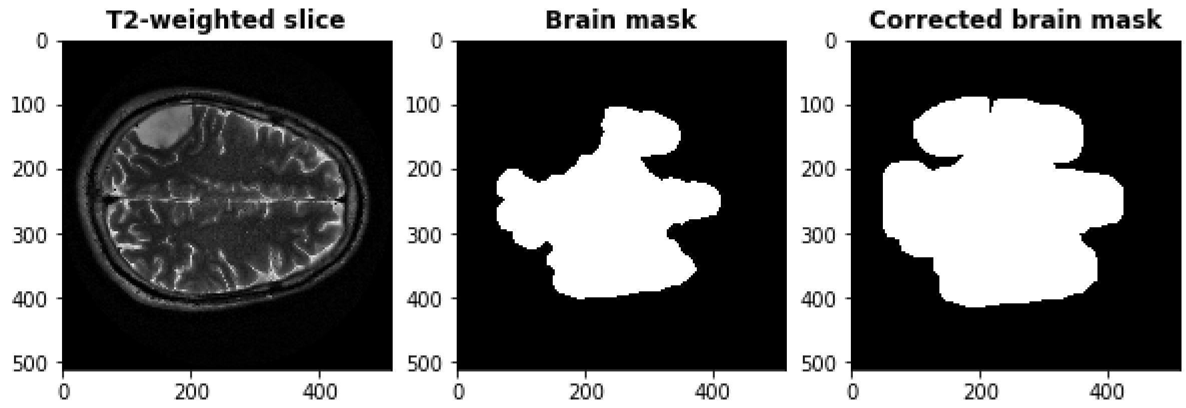


Figure 3.2: Brain mask correction. From left to right: original T2-weighted slice; original brain mask, which clearly misses the part where the tumor is located; corrected brain mask obtained by the addition of the brain mask and the tumor mask. A dilation filter and a fill-in-the-holes filter is applied to smooth the result.

3.2.4. Selection of the MR slices and patch extraction

The MRI scans are 3D files. However, a CNN works with 2D images. Therefore, the 3D files had to be split in 2D slices. Only the slices containing tumor were relevant for this study (the tumor mask was employed to localize them).

The CNN algorithm classifies each slice independently, but we were interested in evaluating the classifier at the patient level. Therefore, a criterion to select the best tumorous slices to ease the classification task was designed. A diagram illustrating the slice selection process using patient BTD-0001 from the EMC/HMC dataset as an example is presented in Figure 3.3.

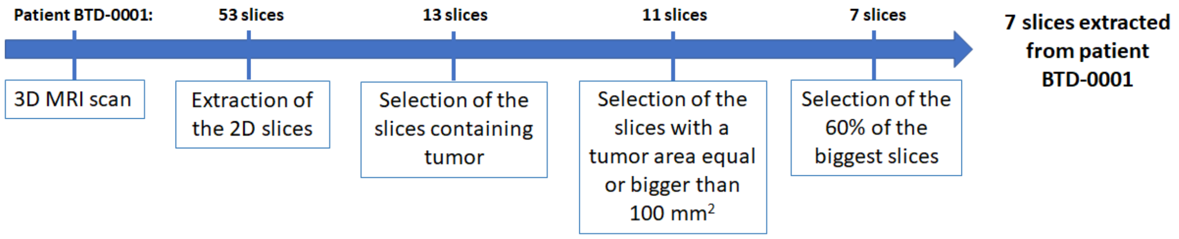

Patient BTD-0001:
53 slices
13 slices
11 slices
7 slices
7 slices extracted from patient BTD-0001
3D MRI scan
Extraction of the 2D slices
Selection of the slices containing tumor
Selection of the slices with a tumor area equal or bigger than 100 mm²
Selection of the 60% of the biggest slices

Figure 3.3: Diagram showing the selection of slices per patient. The 3D MRI scan is split into 53 axial slices, 13 being the ones containing tumor. The area of each tumor is calculated. Tumors having less than 100 mm^2 are discarded, resulting in 11 slices. From this 11 slices, only 60% of the biggest are kept, resulting in 7 slices.

First, we introduced an automatic correction to remove the mislabeled slices during the segmentation procedure. The correction consisted of discarding the slices with an area smaller than $100mm^2$ (the value being decided after studying the tumor area distribution across the slices). Second, we decided to further reduce the amount of slices to ensure that only the biggest slices of tumor were input to the CNN. We assumed that the biggest slices contained most of the relevant information for the classification task. Therefore, we sorted in descending order the areas of each tumor slice and selected the first 60%. The area calculations were done using the bounding box algorithm of the `ITK` library. As shown in Figure 3.3, from patient BTD-0001 we extracted 7 out of the 53 slices contained in the MR scan. Not all the MR scans had the same axial resolution. Therefore, the number of extracted slices varies per patient.

After selecting the relevant slices, we extracted a square patch containing the tumor from each of them. The size of the patch was chosen based on the dimensions of the tumor. With the help of the bounding box algorithm of the `ITK` library, we measured the width and length of each tumor and selected the largest dimension l. The final patch size l_{final} was set as: $l_{final} = l + 0.20l$. However, each tumor has a different size, resulting in different dimensions for every extracted patch. The implementation of the CNN algorithm is easier when working with input images of the same size. Therefore, we resized all the patches to a fixed dimension of 128x128 pixels using the resampling algorithm of the `ITK` library with a linear interpolator. The patch extraction step has been incorporated during the training time in order to ease the data augmentation step.

3.3. Classification algorithm

3.3.1. Network architecture

For this study, we designed our own CNN using the `Keras` library with the `tensorflow` backend. Information from previous radiogenomic studies suggest that both local features (such as texture features) and global features (such as shape) contribute to the prediction of the 1p/19q co-deletion status [33, 35]. Therefore, we designed our network based on this information. In addition, we tried to limit the depth of our CNN to reduce the number of parameters, since our dataset had only 284 patients.

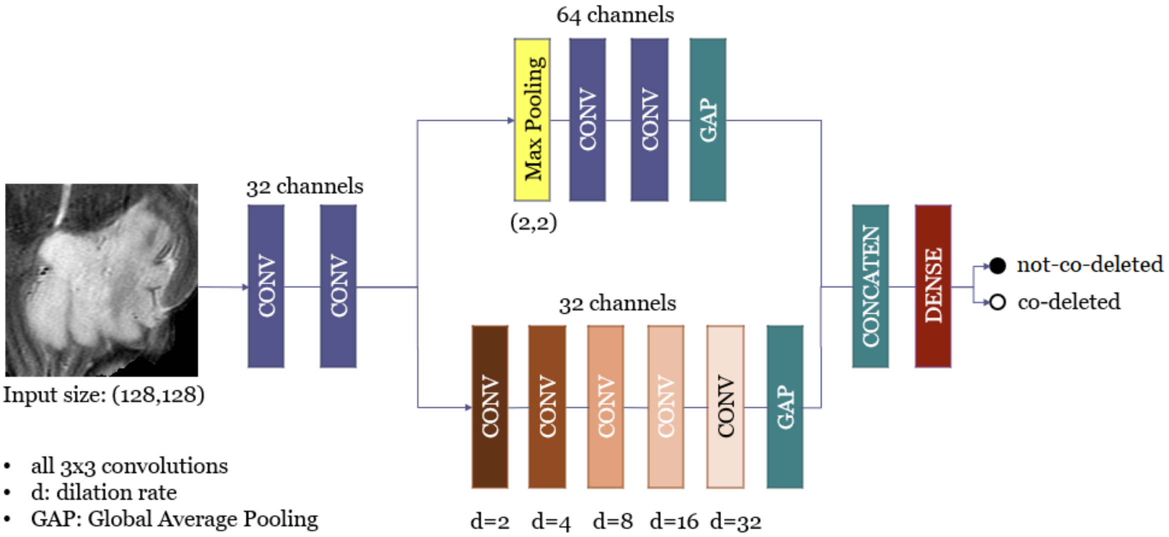


Figure 3.4: Diagram of the CNN, which employs 128x128 images containing the tumors to predict the 1p/19q co-deleted status. The initial common path and the secondary global path (lower branch of the bifurcation) have 32 filters in each of the convolutional modules CONV (further described in Figure 3.5). Dilated convolutions with exponentially increasing dilation rate (d) are employed to extract features in the global path. The secondary local path (upper branch of the bifurcation) has 64 filters in each of the CONV modules. All the CONV modules have 3x3 convolutions kernels. Global average pooling layers (GAP) were employed to extract the feature vectors, which were concatenated using a concatenate layer and finally input to a 2-unit dense layer which outputs the predicted class.

The CNN can be divided in three parts. The initial common path is made out of two convolutional blocks of 32 channels. The output feature maps are then fed into two independent paths, which we named the 'local' path and the 'global' path. The local path includes an initial max pooling layer with a (2,2) stride and is followed by two convolution modules with 64 channels. Global average pooling is used to obtain the local feature vector. The global path is a succession of five dilated convolutions of 32 channels, with a dilation rate that increases following the power of 2 distribution (i.e 2, 4, 8, 16 and 32). Dilated convolutions [60] were chosen to increase the receptive field of the features, the region of the input image from which they are derived, without reducing the resolution of the feature maps. The number of dilated convolutions was chosen so that we had a final receptive field which encompassed the whole 128x128 tumor patch, from which to derive the global features. Once again, a global average pooling layer is employed to obtain the global feature vector. Both feature vectors are concatenated and then input to the last 2-neurons dense layer, which has the softmax activation. Figure 3.4 depicts the scalar architecture.

All the convolutional modules are made out of a 3x3 convolutional layer, followed by a batch normalization layer and a ReLU activation function, as depicted in Figure 3.5. The objective of the batch normalization layer is to reduce the covariance shift and to act as a regularizer. The Adam optimizer is employed to optimize the categorical cross-entropy loss function. The weights were initialized following the heuristics described in He et al. [61] and the bias were initialized with zeros. L2-regularization with a parameter of 0.01 was employed in each convolutional layer to further contribute to the regularization of the network. Hyperparameters were selected based on the Chang et al. [62] paper, whose purpose was also to classify gliomas based on the 1p/19q co-deletion status.

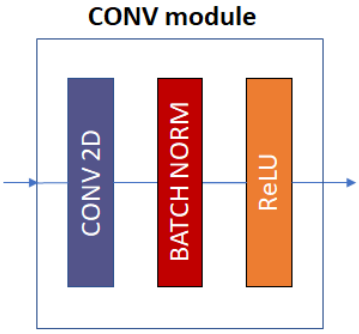


Figure 3.5: Diagram of the convolutional module. The module consists of three layers: a 3x3 2D convolution, a batch normalization layer and an activation layer with a ReLU.

Dynamic data augmentation

Dynamic data augmentation was used during training to increase the variability of the training set and thus to reduce the overfitting of the classifier. It consists of applying mathematical transformations to the images, such as rotations and translations. The code employed is based on the `Keras` data augmentation source code. For our algorithm, we decided to incorporate random rotations within a (-20°,20°) range, random translation in both directions within a (-10%, 10%) range of the dimension of the extracted patch and random left and right flipping according to a Bernouilli distribution. We decided the rotation range according to the expected possible rotations that a patient would move his head in the MR machine. Rotations are performed to the whole image, right before the patch extraction step, and thus a recalculation of the tumor center was required. The translation operation follows the patch extraction. The range was chosen so as to ensure that the full tumor is still contained within the margins of the patch. Finally, we decided to only perform left and right flipping (and not up and down), because the brain structure is quasi-symmetric in this direction, but not in the other.

3.3.2. Implementation of the LBP convolutional module

The LBP module is inserted at the beginning of the CNN to form the LBCNN, depicted in Figure 3.6. It generates the LBP image which is passed as a second channel to the CNN. When working with both T1-weighted and T2-weighted modalities, each input channel has its own LBP module, to create independent LBP images from each modality.

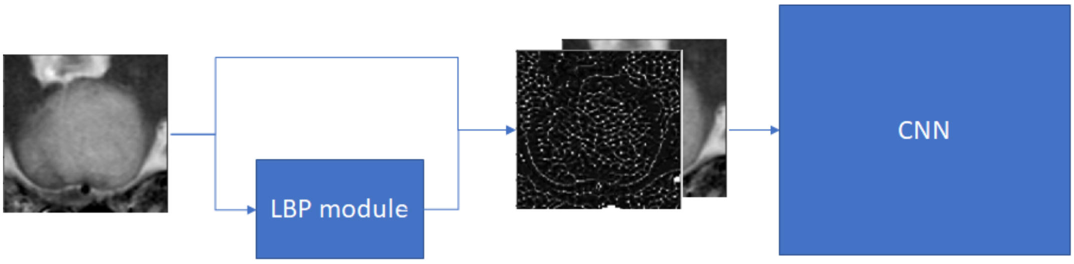


Figure 3.6: Diagram of LBCNN using LBP of radius 1.

The LBP module encompasses five different non-trainable layers: two 2D convolutional layers, two lambda layers and a batch normalization layer as depicted in Figure 3.7.

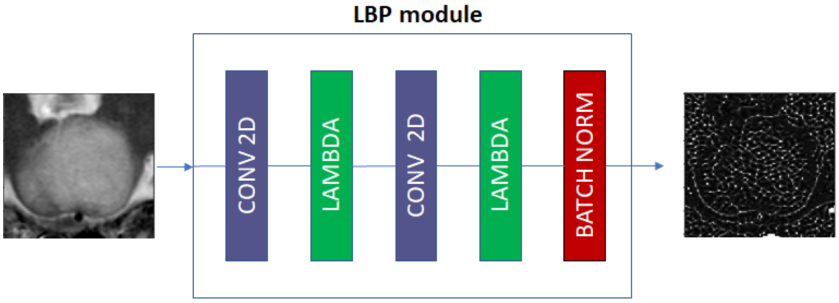


Figure 3.7: LBP convolutional module.

The layers design is explained for radius 1 LBPs. The first 2D convolutional layer creates the eight 3x3x1 sparse filters, as the ones depicted in Figure 2.6, to emulate the difference operation between the central pixel of the kernel and its neighbors, as stated in Section 2.3.1. In the LBP descriptor algorithm, an interpolation operation is used to compute the exact intensity value of the eight points that form the radius 1 circle, but in our study we decided to follow the approximation stated in the paper of Juefi et al. [54] and use the intensity value of the pixels containing the points.

The first lambda layer simulates the threshold operation that creates the bit-map images, by using the function: $f(x) = clip(sign(x)+1,0,1))$ 0 and 1 being the low and high limits of the clipping function $clip(x)$. A bit image requires an intensity range between (0,1), but the range of the $sign(x)$ function is between (-1,1). Therefore, we decided to add 1 to shift the range to (0,2) and then use the $clip(x)$ function to obtain the (0,1) range.

Then, a second 1x1x8 2D convolution layer computes the eight possible LBP maps, depicted in Figure 3.8, each of the eight 1x1x8 kernels being the result of an encoding direction. As explained in Section 2.3.1, each of the LBP maps is constructed by computing the weighted sum of the eight bit map images using the binary weights ($2^0,2^1,2^2,2^3,2^4,2^5,2^6,2^7$), encoded in the weights of each 1x1x8 convolutional filter (i.e the first layer of the first 1x1x8 convolutional filter has a weight of value 2^0 and the last layer has a weight of value 2^7). Each of the 1x1x8 convolutional filters has a specific order of the binary weights, to compute the eight encoding directions. To compute the second 1x1x8 convolutional filter, the values of the weights are shifted by one position (i.e the first layer of the second 1x1x8 convolutional filter has a weight of value 2^1 and the last layer has a weight of value 2^0). The procedure is repeated to obtain the rest of the encoding directions.

A second lambda layer combines the eight LBP maps into one rotational invariant LBP, following the equation stated at Equation 2.6. The final batch normalization layer ensures that the scale of the LBP^{ri} image is similar to the one from the original normalized input. Figure 3.9 depicts the output of the layers of the LBP module.

The construction of higher radius LBP images (i.e radius 3 and radius 5) is practically equivalent to the one described above. The only layer that changes is the first convolutional layer, and more specifically the size of the kernels and the distribution of the values of the sparse weights. We decided to keep the number of filters to eight (and thus the number of points P to represent the LBP operator), to keep the binary range of

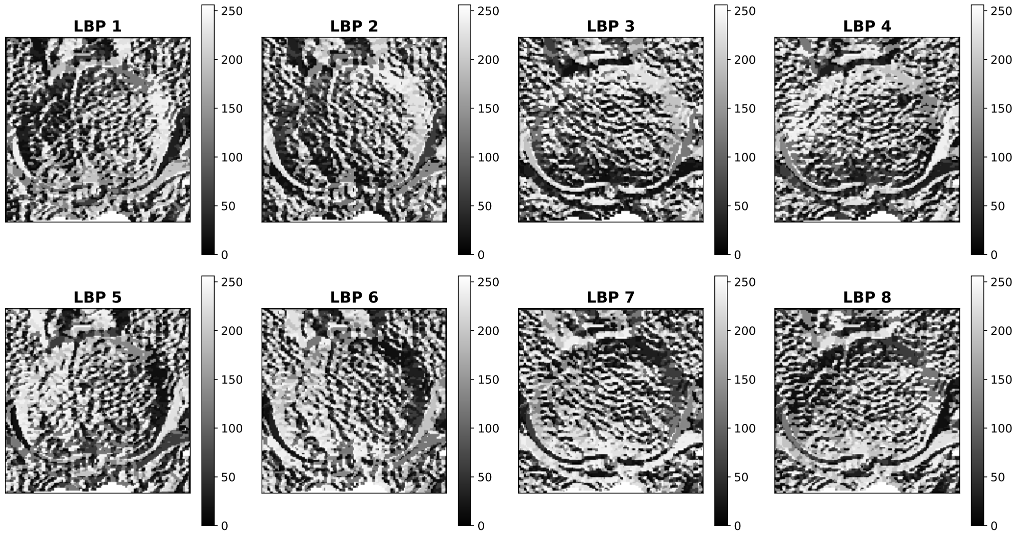


Figure 3.8: LBP maps in the eight encoding directions, highlighting the different directional texture patterns.

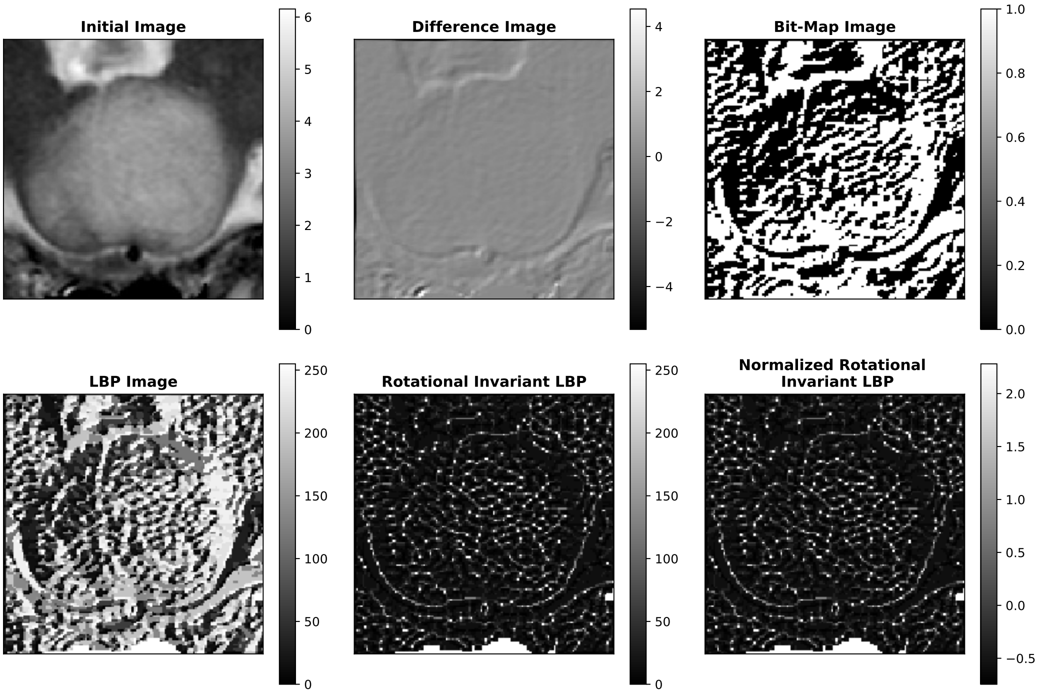
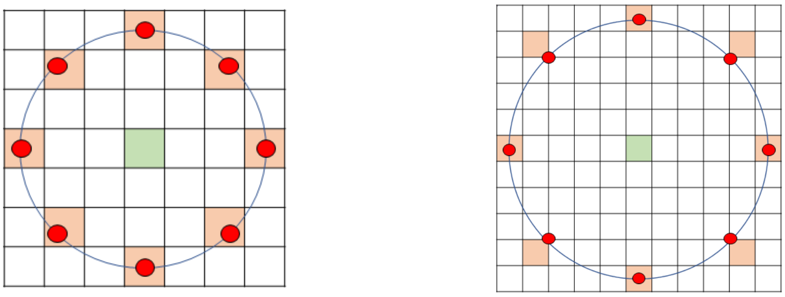


Figure 3.9: Construction of the normalized radius 1 LBP^{ri} image. Top row from left to right: the normalized input image; one of the eight difference maps created from the first convolutional layer; one of the eight bit-maps created after the lambda layer emulating the threshold operation. Bottom row from left to right: one of the eight LBP after the second convolutional layer emulating the weighted sum of the bit-maps with a predefined encoding direction; the LBP^{ri} combining the eight possible LBPs; the normalized LBP^{ri}.

the different radius LBPs and to avoid ill-conditioning the network (further details about this are explained in appendix B). Radius 3 LBPs have convolutional kernels of size 7x7x1 and radius 5 LBPs of size 11x11x1.

To select the non-zero pixels of each of the eight convolutional kernels (which are the ones that are going to be compared to the central pixel of the kernel), we plotted the radius 3 and 5 circles over a 7x7 and 11x11 lattice respectively and equally distributed eight points. The pixels over which the points fell were the selected ones.



(a) Latice for LBP of radius 3.

(b) Lattice for LBP of radius 5

Figure 3.10: Patterns to select the neighbor pixels which are going to be compared to the central pixel of the kernel to create the LBP images of radius 3 and 5. A 7x7 and 11x11 lattice are used respectively. The red circles correspond to the points which are compared to the central pixel to form the 8 points LBPs. The orange pixels indicate the position at which the red points have fallen. The green pixel corresponds to the center of the lattice.

Figure 3.10 depicts the result of the process. For the case of the radius 5 LBP, the points fell exactly on the edge of four pixels. We decided to take the pixel closer to the edges of the lattice, to increase the scope of the LBP. Figure 3.11 shows the output of the different radius LBPs.

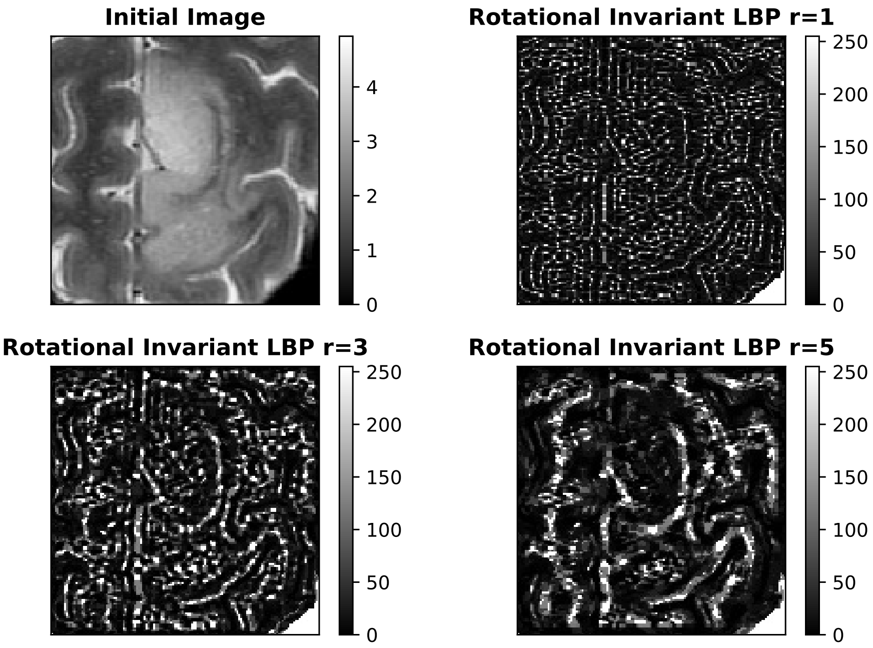

Initial Image
4
3
2
1
0
Rotational Invariant LBP r=1
250
200
150
100
50
0
Rotational Invariant LBP r=3
250
200
150
100
50
0
Rotational Invariant LBP r=5
250
200
150
100
50
0

Figure 3.11: LBP images of radius (r) 1 (top right), 3 (bottom left) and 5 (bottom right). The input image is a T2-weighted patch of a LGG. By increasing the radius, the scale of the patterns obtained increases.

3.3.3. Experiments overview

A total of five architectures were trained using a 5-fold stratified cross-validation scheme randomly created from the EMC/HMC dataset (the details being explained in Appendix C):

- The designed CNN,
- The designed LBCNN with LBP of radius 1 (LBCNN1);
- The designed LBCNN with LBP of radius 3 (LBCNN3);

- The designed LBCNN with LBP of radius 5 (LBCNN5);
- The designed LBCNN with the combination of the three LBPs of radius 1, 3 and 5 (LBCNN$_{all}$).

Each algorithm is trained twice from scratch during 300 epochs, one time with T2-weighted images only and a second time using both T2-weighted and T1-weighted images. To select the best model out of the 300 epochs, we selected the one with the highest F1-score metric per patient on the validation set. The metrics per patient were created by computing the median of the probabilities of all the slices belonging to each patient of the validation set. We computed the F1-score, the ROC-AUC, the precision, the specificity, the sensitivity and the accuracy metrics.

Each architecture was evaluated on the TCIA dataset to further validate the algorithm on a completely independent dataset. For that purpose, we created an ensemble classifier out of the best model derived during the training of each of the 5 folds. The ensemble probabilities per slice were computed using the median of the probabilities of each fold. The same metrics per patient were obtained in the same way as mentioned above.

To validate the design of the CNN, we compared our performance metrics with the ones of an SVM classifier which trained on the same exact dataset (the paper from which we extracted the results is not published yet, but it follows the approach explained in van der Voort et al. [32]). The SVM was trained over a 500-fold cross-validation scheme using both T1-weighted and T2-weighted images. In this case, we reported the mean values of the 5-fold cross-validation together with the confidence intervals calculated following the paper by Nadeau et al. [63].

To validate the design of the LBP module, we created the LBP images in the pre-processing step with the $scikit-image$ library and directly give them as a second channel in the CNN. For that purpose, we only used the T2-weighted images and the radius 5 LBP. We kept the number of points to 8 to further evaluate our filters design, shown in Figure 3.10b, which was an approximation of the real LBP descriptor (same training and evaluation procedures as described above were used).

Finally, to evaluate the effect of working with uniform LBPs, which were the ones employed in the SVM classifier, we trained the CNN giving uniform LBPs as a second channel. Once again, we only used the T2-weighted images and the radius 5 LBP. However, this time we increased the resolution to 24 points, to exactly replicate the LBPs used in the aforementioned unpublished SVM study (same training and evaluation procedures as described above were used).

4

Results

4.1. Performance metrics of the classifiers

4.1.1. CNN classifier

The performance metrics of the CNN classifier on both the EMC/HMC (training) and TCIA (test) datasets are presented in Table 4.1.

Table 4.1: Performance metrics of the CNN classifier on both the EMC/HMC training dataset and the TCIA test dataset. Training results are represented using the mean and standard deviation (mean ± standard deviation) and test results using the value of the ensemble classifier. Metrics for both T2-weighted images only (T2w) and T2-weighted and T1-weighted (T2w + T1w) combined are shown.

CNN	**T2w**		**T2w + T1w**	
	EMC/HMC	**TCIA**	**EMC/HMC**	**TCIA**
F1-score	0.767 ± 0.066	0.676	0.752 ± 0.060	0.765
AUC	0.842 ± 0.074	0.832	0.821 ± 0.077	0.841
Precision	0.737 ± 0.059	0.842	0.676 ± 0.119	0.891
Specificity	0,841 ± 0.045	0.795	0.744 ± 0.157	0.841
Sensitivity	0.803 ± 0.119	0.565	0.858 ± 0.065	0.671
Accuracy	0.830 ± 0.040	0.643	0.792 ± 0.082	0.729

T2-weighted experiments

In general, the training set reports better metrics than the test set (especially in terms of sensitivity). However, the precision of the test set is higher than the mean value of the training set. The AUC metric remains almost the same in both training and test set.

T1-weighted and T2-weighted experiments

In general, the test set reports better metrics than the training set (especially in terms of precision and specificity). However, these two metrics on the training set also have a standard deviation higher than than 10%. In addition, the training set has a higher sensitivity than the test set. The values of the F1-score and the AUC of the test set are similar to the ones of the mean of the training set.

T2-weighted input vs T1-weighted and T2-weighted inputs

The metrics of the combined modalities classifier, except from the sensitivity, are lower than the combined modalities classifiers on the training set. It also reports higher standard deviations (except for the sensitivity).

The metrics of the combined modalities classifier on the test set are all higher than the ones of the single modality classifier, especially the F1-score, the sensitivity and the accuracy metrics.

4.1.2. LBCNN1 classifier

The performance metrics of the LBCNN1 classifier on both the EMC/HMC (training) and TCIA (test) datasets are presented in Table 4.2.

Table 4.2: Performance metrics of the LBCNN1 classifier on both the EMC/HMC training dataset and the TCIA test dataset. Training results are represented using the mean and standard deviation (mean ± standard deviation) and test results using the value of the ensemble classifier. Metrics for both T2-weighted images only (T2w) and T2-weighted and T1-weighted (T2w + T1w) combined are shown.

LBCNN1	**T2w**		**T2w + T1w**	
	EMC/HMC	**TCIA**	**EMC/HMC**	**TCIA**
F1-score	0.755 ± 0.064	0.713	0.731 ± 0.073	0.790
AUC	0.851 ± 0.071	0.821	0.828 ± 0.082	0.819
Precision	0.738 ± 0.082	0.879	0.664 ± 0.112	0.831
Specificity	0.846 ± 0.062	0.841	0.759 ± 0.121	0.705
Sensitivity	0.776 ± 0.084	0.600	0.818 ± 0.057	0.753
Accuracy	0.823 ± 0.047	0.682	0.782 ± 0.078	0.736

T2-weighted experiments

In general, the training set reports better metrics than the test set (especially in terms of sensitivity). However, the precision on the test set is higher than the mean value of the training set (0.879 and 0.738 respectively). The specificity and the AUC metrics are similar in both test and training set.

T1-weighted and T2-weighted experiments

In general, the metrics of the test set are more balanced that the ones of the training set. The test set has a higher precision than the training set. However, the specificity and especially the sensitivity are higher in the training set than in the test set. Both precision and specificity have a standard deviation higher than 10% in the training set. The AUC metric remains almost the same.

T2-weighted input vs T1-weighted and T2-weighted inputs

Regarding the training set, except for the sensitivity, the mean values of the combined modalities classifier are lower than the single modality classifier. In addition, the former reports higher standard deviations.

However, the metrics of the combined modalities classifier on the test set are more balanced that the ones of the single modality. The former has higher sensitivity and accuracy but lower specificity and precision.

4.1.3. LBCNN3 classifier

The performance metrics of the LBCNN3 classifier on both the EMC/HMC (training) and TCIA (test) datasets are presented in Table 4.3.

T2-weighted experiments

The classifier on the test set has better specificity and especially better precision than the training set. However, it has a lower sensitivity and accuracy than the training set. The F1-score and the AUC metrics are similar on both the training and test set.

T1-weighted and T2-weighted experiments

In general, the metrics of the test set are better than the ones of the training set. The classifier on the test set has better F1-score, AUC and especially it has better precision. However, it has a lower sensitivity and accuracy than the test set.

Table 4.3: Performance metrics of the LBCNN3 classifier on both the EMC/HMC training dataset and the TCIA test dataset. Training results are represented using the mean and standard deviation (mean ± standard deviation) and test results using the value of the ensemble classifier. Metrics for both T2-weighted images only (T2w) and T2-weighted and T1-weighted (T2w + T1w) combined are shown.

LBCNN3	**T2w**		**T2w + T1w**	
	EMC/HMC	**TCIA**	**EMC/HMC**	**TCIA**
F1-score	0.752 ± 0.060	0.748	0.749 ± 0.060	0.810
AUC	0.840 ± 0.051	0.833	0.847 ± 0.044	0.872
Precision	0.688 ± 0.084	0.887	0.688 ± 0.073	0.877
Specificity	0.791 ± 0.077	0.841	0.791 ± 0.070	0.795
Sensitivity	0.830 ± 0.027	0.647	0.826 ± 0.091	0.753
Accuracy	0.805 ± 0.058	0.713	0.805 ± 0.049	0.767

T2-weighted input vs T1-weighted and T2-weighted inputs

The mean values of the performance metrics on the training set are the same for both combined modalities and single modalities classifiers. In addition, the combined modalities classifier has lower standard deviations in all the metrics except from the sensitivity.

If we compare the single and combined modalities experiments on the test set, we can observe than in general, the combined modalities have higher metrics. It is only lower on the specificity.

4.1.4. LBCNN5 classifier

The performance metrics of the LBCNN5 classifier on both the EMC/HMC (training) and TCIA (test) datasets are presented in Table 4.4.

Table 4.4: Performance metrics of the LBCNN5 classifier on both the EMC/HMC training dataset and the TCIA test dataset. Training results are represented using the mean and standard deviation (mean ± standard deviation) and test results using the value of the ensemble classifier. Metrics for both T2-weighted images only (T2w) and T2-weighted and T1-weighted (T2w + T1w) combined are shown.

LBCNN5	**T2w**		**T2w + T1w**	
	EMC/HMC	**TCIA**	**EMC/HMC**	**TCIA**
F1-score	0.755 ± 0.046	0.771	0.742 ± 0.041	0.800
AUC	0.852 ± 0.064	0.861	0.866 ± 0.053	0.833
Precision	0.678 ± 0.078	0.868	0.681 ± 0.063	0.825
Specificity	0.768 ± 0.092	0.795	0.785 ± 0.069	0.682
Sensitivity	0.857 ± 0.082	0.694	0.818 ± 0.057	0.776
Accuracy	0.802 ± 0.050	0.729	0.798 ± 0.041	0.744

T2-weighted experiments

The classifier on the test set has better precision than on the training set. However, it has lower sensitivity and accuracy. The remaining metrics are similar in both training and test set.

T1-weighted and T2-weighted experiments

The training set has higher specificity and sensitivity values than the test set. However, the test set has better precision and f1-score than the training set. The AUC is similar on both training and test set.

T2-weighted input vs T1-weighted and T2-weighted inputs

The metrics on the training set are very similar for both single and combined modalities classifiers. However, the combined classifier has lower standard deviations for all the metrics.

Regarding the test set, the single modality classifier has better precision, specificity and AUC than the combined modalities classifier. However, it has a lower sensitivity and F1-score than the combined modalities classifier.

4.1.5. LBCNN$_{all}$ classifier

The performance metrics of the LBCNN$_{all}$ classifier on both the EMC/HMC (training) and TCIA (test) datasets are presented in Table 4.5.

Table 4.5: Performance metrics of the LBCNN$_{all}$ classifier on both the EMC/HMC training dataset and the TCIA test dataset. Training results are represented using the mean and standard deviation (mean ± standard deviation) and test results using the value of the ensemble classifier. Metrics for both T2-weighted images only (T2w) and T2-weighted and T1-weighted (T2w + T1w) combined are shown.

LBCNN$_{all}$	**T2w**		**T2w + T1w**	
	EMC/HMC	**TCIA**	**EMC/HMC**	**TCIA**
F1-score	0.770 ± 0.059	0.762	0.751 ± 0.043	0.759
AUC	0.851 ± 0.088	0.868	0.859 ± 0.058	0.816
Precision	0.740 ± 0.129	0.903	0.694 ± 0.036	0.822
Specificity	0.833 ± 0.099	0.864	0.804 ± 0.024	0.705
Sensitivity	0.809 ± 0.042	0.659	0.818 ± 0.057	0.706
Accuracy	0.826 ± 0.058	0.729	0.809 ± 0.032	0.705

T2-weighted experiments

The test set has higher specifity and especially a higher precision than the training set. However, the standard deviations of the these metrics on the training set reach 10%. In addition, the sensitivity and accuracy on the test set are lower than in the training set. The F1-score and AUC metrics remain similar in both training and test set.

T1-weighted and T2-weighted experiments

Except for precision, which is higher in the test set, the rest of the metrics are higher in the training set than in the test set (the only similar value between the test set and the training set is the F1-score.).

T2-weighted input vs T1-weighted and T2-weighted inputs

The performance metrics of both single and combined modalities classifiers are similar on the training set. Only precision and specificity are a bit higher in the single modality classifier. However, the reported standard deviations of the combined modalities classifier are all lower than the single modalitity classifier (except from the sensitivity).

If we compare the results of the metrics on the test set of both single and combined modalities classifiers, in general the single modality classifier has higher metrics than the combined one. Only the sensitivity is better in the combined classifier than in the single modality one.

4.2. Comparison between classifiers

4.2.1. Training set

The performance metrics of the five studied classifiers on the EMC/HMC dataset are compared in Table 4.6. Results are only shown for the combined imaging modalities.

In general, the LBCNN classifiers do not show better mean performance metrics than the CNN classifier. In fact, all the performance metrics show very similar mean values for the five architectures. However, the LBCNN5 classifier and especially the LBCNN$_{all}$ classifier report lower standard deviations in all the metrics than the CNN classifier, the standard deviation of the LBCNN$_{all}$ classifier being lower than 6%. The CNN and the LBCNN1 classifiers report standard deviations on higher than 6% and exceeding 10% in both precision and specificity.

Table 4.6: Performance metrics of the five studied architectures on the EMC/HMC dataset using T2-weighted and T1-weighted images (mean ± standard deviation).

	CNN	LBCNN1	LBCNN3	LBCNN5	LBCNN$_{all}$
F1-score	0.752 ± 0.060	0.731 ± 0.073	0.749 ± 0.060	0.742 ± 0.041	0.751 ± 0.043
AUC	0.821 ± 0.077	0.828 ± 0.082	0.847 ± 0.044	0.866 ± 0.053	0.859 ± 0.058
Precision	0.676 ± 0.119	0.664 ± 0.112	0.688 ± 0.073	0.681 ± 0.063	0.694 ± 0.036
Specificity	0.744 ± 0.157	0.759 ± 0.121	0.791 ± 0.070	0.785 ± 0.069	0.804 ± 0.024
Sensitivity	0.858 ± 0.065	0.818 ± 0.057	0.826 ± 0.091	0.818 ± 0.057	0.818 ± 0.057
Accuracy	0.792 ± 0.082	0.782 ± 0.078	0.805 ± 0.049	0.798 ± 0.041	0.809 ± 0.032

4.2.2. Independent test set

The performance metrics of the five studied classifiers on the TCIA dataset are compared in Table 4.7. Results are only shown for the combined imaging modalities.

Table 4.7: Performance metrics of the five studied architectures on the TCIA dataset using T2-weighted and T1-weighted images. The values are the result of the ensemble classifier.

	CNN	LBCNN1	LBCNN3	LBCNN5	LBCNN$_{all}$
F1-score	0.765	0.790	0.810	0.800	0.759
AUC	0.841	0.819	0.872	0.833	0.816
Precision	0.891	0.831	0.877	0.825	0.822
Specificity	0.841	0.705	0.795	0.682	0.705
Sensitivity	0.671	0.753	0.753	0.776	0.706
Accuracy	0.729	0.736	0.767	0.744	0.705

All the classifiers have AUC values higher than 0.800, the LBCNN3 classifier having the highest value (0.877). In addition, all the classifiers have precision values higher than 0.800, the CNN classifier having the highest value (0.891). The F1-score value is also higher than 0.750 for all the classifiers, the LBCNN3 having the highest value (0.810). In general, all the classifiers have good performance metrics. However, if we look closer, we can see some differences.

Despite having the highest precision and specificity, the CNN classifiers has the lowest sensitivity (0.671). The LBCNN5 classifier has the highest sensitivity value (0.776) but the lowest specificity (0.682). The LBCNN3 classifiers is the algorithm with the most balanced metrics, reporting precision and sensitivity values similar to the CNN and LBCNN5 classifier respectively.

4.3. Comparison of the SVM with the CNN

Table 4.8 shows the performance metrics of the SVM and the CNN classifiers on the T2-weighted images of the EMC/HCM dataset after the cross-validation approach (500 folds for the SVM and 5 folds for the CNN; the SVM results are extracted from the unpublished continuation of the work done in van der Voort et al. [32]). The mean values of all the metrics are higher for the CNN than for the SVM. However, the confidence intervals of the CNN classifier are in general wider than the ones from the SVM. In some cases we even reach values below the random guess (for the specificity and the precision), and above 1 (for the specificity).

4.4. Effect of using pre-computed LBPs

4.4.1. Validation of the LBP module

Table 4.9 compares the performance metrics of the LBCNN5 classifier with the ones of the CNN classifier having pre-computed, rotational invariant, radius 5 LBP images as a second channel. Both classifiers are tested on the T2-weighted modality only and on both EMC/HMC and TCIA datasets.

Table 4.8: Performance metrics of the SVM and CNN approaches on the EMC/HMC dataset. Results are represented using the mean and 95% confidence interval (mean (95% CI)). Metrics are reported only for the combined T1-weighted and T2-weighted images.

	SVM (mean (95%CI))	**CNN (mean (95%CI))**
F1-score	0.701 (0.640 - 0.761)	0.752 (0.642 - 0.866)
AUC	0.755 (0.694 - 0.817)	0.821 (0.680 - 0.968)
Precision	0.570 (0.491 - 0.649)	0.676 (0.464 - 0.906)
Specificity	0.721 (0.628 - 0.813)	0.744 (0.468 - 1.051)
Sensitivity	0.657 (0.562 - 0.752)	0.858 (0.738 - 0.982)
Accuracy	0.698 (0.636 - 0.760)	0.792 (0.642 - 0.949)

Table 4.9: Performance metrics of the CNN classifier with pre-computed LBP5 images and the LBCNN5 classifier on the EMC/HMC and TCIA datasets. Training results are represented using the mean and standard deviation (mean ± standard deviation) and test results using the value of the ensemble classifier. Metrics are only reported for T2-weighted images.

	LBCNN5		**CNN + LBP5^{ri}**	
	EMC/HMC	**TCIA**	**EMC/HMC**	**TCIA**
F1-score	0.755 ± 0.046	0.771	0.755 ± 0.056	0.815
AUC	0.852 ± 0.064	0.861	0.833 ± 0.074	0.870
Precision	0.678 ± 0.078	0.868	0.693 ± 0.112	0.889
Specificity	0.768 ± 0.092	0.795	0.783 ± 0.110	0.818
Sensitivity	0.857 ± 0.082	0.694	0.837 ± 0.082	0.753
Accuracy	0.802 ± 0.050	0.729	0.805 ± 0.059	0.775

Overall, the mean values of the performance metrics of both classifiers on the training set are very similar. However, the standard deviations of the LBCNN5 classifier are lower than the ones of the CNN with the rotational invariant LBPs. Regarding the test set, the classifier with the rotational invariant LBPs has higher performance metrics than the LBCNN5 classifier, especially in terms of sensitivity.

4.4.2. Using uniform LBPs

Table 4.10 compares the performance metrics of the CNN classifier using pre-computed rotational invariant radius 5 LBP images as a second channel and pre-computed uniform radius 5 LBP images as a second channel.

Table 4.10: Performance metrics of the CNN classifier with pre-computed rotational invariant and uniform radius 5 LBP images on the EMC/HMC and TCIA datasets. Training results are represented using the mean and standard deviation (mean ± standard deviation) and test results using the value of the ensemble classifier. Metrics are only reported for T2-weighted images.

	CNN + rorLBP5		**CNN + uniLBP5**	
	EMC/HMC	**TCIA**	**EMC/HMC**	**TCIA**
F1-score	0.755 ± 0.056	0.815	0.747 ± 0.038	0.790
AUC	0.833 ± 0.074	0.870	0.843 ± 0.053	0.819
Precision	0.693 ± 0.112	0.889	0.706 ±0.064	0.831
Specificity	0.783 ± 0.110	0.818	0.813 ± 0.064	0.705
Sensitivity	0.837 ± 0.082	0.753	0.797 ± 0.079	0.753
Accuracy	0.805 ± 0.059	0.775	0.809 ± 0.031	0.736

Overall, the mean values of the performance metrics of both classifiers on the training set are very similar. However, the standard deviations of the CNN classifier with the uniform LBPs are lower than the ones of the CNN with the rotational invariant LBPs. Regarding the test set, the classifier with the rotational invariant LBPs has higher performance metrics than the classifier with the uniform LBPs.

5

Conclusion and Discussion

5.1. Analysis of the results

This project studied the effect of using local binary convolutional neural networks to non-invasively predict the 1p/19q co-deletion status of presumed low grade gliomas from pre-operative MRI scans. Our results show that LBCNNs are able to predict the 1p/19q co-deletion status. In general, the performance metrics of the combined T2-weighted and T1-weighted classifiers are higher than the ones of the single modality classifier, as also proved in van der Voort et al. [32]. However, LBCNNs do not improve the performance metrics of the default CNN classifier on the training set, since the reported metrics of all the architectures were very similar (as shown in Table 4.6). Larger radius LBCNNs though have lower variance when using different training data, suggesting that the LBP features contribute to the robustness of the classifier. By comparing the results on the EMC/HMC dataset with the ones of the TCIA dataset (see Tables 4.6 and 4.7), it can be seen that LBCNNs are robust in front of unseen data. However, the default CNN classifier is also robust in front of unseen data. This suggests that the default CNN can already extract similar LBP features from the images. However, LBCNNs classifiers, especially the LBCNN3 classifier, help increasing the sensitivity while minimally reducing precision.

Results from Table 4.9 show that our approximation of the higher radius LBPs captures most of the information extracted from the pre-computed LBPs^{ri}. Regarding the pre-computed uniform LBPs, results from table 4.10 imply that working with pre-computed uniform LBPs does not improve the results obtained with LBPs^{ri}. Therefore, further research in designing convolutional modules to emulate the uniform LBP images is not required.

However, pre-computed LBPs^{ri} reported lower standard deviations, suggesting that they increase the robustness of the classifier when using different input data. In addition, pre-computed LBPs^{ri} report higher metrics on the test set than the LBCNN5 classifier, implying that the default CNN with pre-computed LBPs^{ri} adapts better to unseen data. Still, having the LBP module slightly speeds up the time that it takes to train the algorithm (120 ms per epoch against 130 ms per epoch with LBP module and with pre-computed LBPs^{ri} respectively). Results imply that the LBP module does not add further information than the pre-computed LBPs^{ri} images.

Initially, the purpose of including the LBP module as a part of the CNN was to learn the optimal resolution and scale for the LBP descriptor, and thus the optimal size of the convolutional kernels and the weight distribution required to predict the 1p/19q co-deletion status. However, to overcome design architecture difficulties and to reduce the number of studied textural scales, in the end we preferred to validate the findings of the unpublished study of van der Voort et al. which suggested that radius 5 uniform LBP contributed to the prediction of the 1p/19q co-deletion of LGG in MRI scans. Nevertheless, we decided to keep the LBP as part of the CNN to set the path for future network architecture designs able to learn the kernel size and the weight distribution of the optimal resolution and scale LBPs.

5.2. Clinical interpretation

The presented study is one of the first in trying to validate deep learning classifiers on an independent dataset, proving its robustness against unseen data. There is only one study that tried to validate their results on an independent dataset (Lu et al. [35]), but using only five patients. Regarding the default CNN algorithm, we obtained a classifier with similar metrics on the training and on the test set. In fact, we reported a precision metric on the test set of 0.890. However, the sensitivity was only 0.670. From these values, we can say that our classifier does not predict a high amount of true positives, but it predicts them with a high precision. Such results might be the consequence of training and testing the classifier with imbalanced dataset with opposed minority class, but further analysis is required to confirm this hypothesis.

Clinically, such a classifier has the preferred balance of performance metrics. As mentioned in the Section 1.2, co-deleted tumors, represented by the positive class in this study, have a better prognosis than not-co-deleted tumors, represented by the negative class. Therefore, a clinician might prefer to wait and watch the evolution of the tumor if it is co-deleted rather than taking the risks that treatment entails. Thus, it is more important for a clinician to be sure that the predicted positive classes are truly co-deleted tumors, than to have a high number of positive predicted classes. The latter scenario could lead to having a high number of false positives, resulting in having not-co-deleted tumors considered as co-deleted which could follow the wait and watch procedure. Such a situation is very risky for a patient that really needs the treatment.

5.3. Limitations and future work of the study

The presented study has some limitations. First of all, the cross-validation was done with five folds only, since training a single classifier can take between half a day to one day. Therefore, despite reporting similar metrics than a SVM classifier (as seen in table 4.8) the calculated confidence intervals for the CNN are not significant. A higher number of folds is required to validate the findings of the study.

In addition, our study employs segmentation masks to extract the tumor patch. Obtaining the segmentation masks is a time consuming task, and the process is prone to inter-observer variability. The latter issue in our study is less problematic, since the segmentation masks are used to localize the tumor rather to set the region of interest from which to extract features, like in the classic machine learning approaches [26, 41, 64]. To remove the need of using segmentation masks and thus to speed up the pipeline, a future step for this research would be to train the CNN with the whole brain slice. Ideally, the CNN should not only learn to extract the relevant features, but also to localize the tumor.

In our study, we decided to work with a 2D CNN approach instead of using a 3D approach to reduce the computational time during training. Clinicians in fact assess the MRI images by scrolling through the 2D slices. However, our approach uses patches of the tumor, resulting in the loss information of the localization of the tumor in the brain. Such a feature has been proven to be very important in the distinction of the 1p/19q co-deleted tumors [4, 33, 35]. Moreover, after extracting the patch, we are resizing it to a fixed dimension, losing information about the size. Working with complete brain slices could overcome the loss of information about the size and localization of the tumor. In addition, clinicians use semantic features, such as age and gender, to decide the next step in the risk-assessment of the patient [5]. Study the effect of adding such features could be a promising step to continue this research project.

The aforementioned clinical argument was also used to select the performance metric on which to focus to select the best model out of the 300 trained epochs. Our goal was to minimize the number of false positives while maximizing the number of true positives (so having a high precision and a high sensitivity). Since the F1-score is the harmonic mean of these two quantities, it was considered as the most informative metric. However, we are basing our assumptions on the metrics reported on the validation set during the training phase, which only has 57 patients. Small validation sets may report higher variability on their metrics, as suggested by the reported values of the standard deviations. In addition, results reported in Appendix D suggest that there might be no direct correlation between the best classifier of such a small validation set and the independent test set. In fact, the maximum value of the f1-score of the test set does not correspond to the one of the validation set. Therefore, gathering a higher number of patients for the study would increase the size of the validation set, reducing the variability of the results and increasing the correlation between the validation and test set.

Moreover, in our study, we are only working with two MRI modalities. Several studies have proven the utility of more advanced MRI modalities such as T2-FLAIR, perfusion and diffuse weighted MRI to extract information from tumors [33, 34, 65, 66]. Therefore, working with more sophisticated MRI modalities could increase the performance metrics of the classifier.

In a similar vein, future work which can be done to increase the performance of the classifier without increasing the size of the dataset could be to train with all the patients, including the ones left for the validation set (which are not the ones from the independent test set), instead of building an ensemble classifier. In fact, a first study depicted in Appendix E suggests that training with all the patients has better performance that the ensemble classifier on the test set (the CNN architecture and T2-weighted images are employed). However, further research is required to choose the criterion to select the best model during the training procedure, to reduce the effect of overfitting on the training set.

The selection of the hyperparameters of the CNN was based on studies pursuing a similar task (Chang et al. [62]), but the network architectures employed were very different. Therefore, another future step in this research would be to optimize our hyperparameters, such as the learning rate, the regularization value and the batch size. In addition, we used sample weights to counteract the effect of the class imbalance in the training set but no in-depth study of the optimal weights has been undertaken (due to the high computational time that the training process takes). Therefore, sample weights are another hyperparameter that require optimization.

Finally, a deeper study on the stochasticity of the network is required. Every time a network is trained from scratch, the weights of the CNN are initialized with different random parameters. A small study about the effect of the random initialization was done by training four different times the CNN with the same input data (reported in Appendix F). Still, a higher number of repetitions is required to obtain significant results.

Acknowledgements

I would like to thank Prof. dr. Wiro Niessen for giving me the opportunity of working in his research group, the Biomedical Imaging Group of Rotterdam (BIGR), at the Erasmus MC hospital, in which I was able to learn the importance of the collaboration between the technical and medical fields. I would also like to thank Dr. ir. Stefan Klein for his insight and guidance during our biweekly meetings. Moreover, I would like to specially thank MSc. Sebastian van der Voort, who has followed and guided my project day by day, providing with all the help that I required every timeI got stuck with my work, and for organizing such amazing board game nights at the hospital which made my experience unforgettable. I would like to thank all the people that I have met at BIGR, which whom I have shared an amazing and super fun time. Finally, I would like to thank all my Delft friends, who had make me feel like home during these past two years.

A

Acquisition parameters

This table has been integrally replicated from the unpublished paper of van der Voort et al., which is the continuation of the work done in van der Voort et al. [32].

	Training dataset (EMC/HMC)		Testing dataset (TCIA)	
MR Setting	T1 min-max	T2 min-max	T1 min-max	T2 min-max
Voxel spacing in-plane (mm)	0.38 x 0.38 - 1.13 x 1.13	0.23 x 0.23 - 1.02 x 1.02	0.47 x 0.47 - 1.1 x 1.1	0.43 x 0.43 - 1.1 x 1.1
Matrix Size	256 x 176 - 1024 x 307	256 x 224 - 1024 x 1024	256 x 256 - 512x512	256 x 256 - 512x512
Echo Time (ms)	1.7 - 20	79.2 - 379	2.6 - 21	12.3 - 108.3
Repetition Time (ms)	3.8 - 1940	2000 - 13468.5	8.2 - 983.3	2033.3 - 8116.6
Slice Thickness (mm)	0.9 - 7.2	1 - 7.2	1 - 5	2 - 5
Slices	19 - 248	19 - 304	20 - 196	20 - 84
Field Strength (Tesla)	0.5, 1.5 or 3.0	0.5, 1.5 or 3.0	1.5 or 3	1.5 or 3

B

Construction criteria for higher order LBPs

The number of sparse convolutional kernels depends on the number of the points P we want to use to construct the LBPs. A higher number of points (and thus filters) implies a higher number of comparisons between voxels and thus better resolution. However, a higher number of filters also increases the amount of binary weights, and thus the range of the LBP feature. The number of points employed to construct the radius 3 and 5 LBP features in the SVM classifier from which we observed the importance of the LBP descriptor in predicting the 1p/19q co-deletion status was 24. The value outputted by 2^{23}, the highest binary weight in this case, is around $8x10^6$. Having such a high value in a pixel of our feature map would probably ill-condition the learning process of the network. The use of the batch normalization layer would probably contribute to reduce the scale of the LBP and to avoid propagating high intensity values into the neural network. However, we would lose the contributions of the smaller LBP values (being almost 0 when normalizing), getting further away from the real LBP image that we want to replicate. Therefore, to keep the range between radius 1, 3 and 5 LBPs and to avoid the aforementioned problem, we decided to keep the resolution of the higher order LBPs to 8 points.

C

Construction of the training and validation datasets

The EMC/HMC dataset was used to train the CNN algorithm, containing 184 not-co-deleted patients (class 0) and 100 co-deleted patients (class 1). Five different training and validation splits were created for cross-validation purposes. To create one of the splits, a stratified approach is employed. The patients are first grouped per class, and then each class group is divided into 5 folds. One fold from each class is used to create the validation set. Thus, the remaining eight folds, four from each class, are used to create the training set. At least one fold from each class is contained once in the validation set of one of the splits.

However, the classification algorithm trains based on slices and not per patient. Therefore, despite the effort employed to create equally distributed splits, the class distribution varies in each split (since we do not extract the same amount of slices per patients, as mention in the previous section). Table C.1 provides with a detailed description of the final class distribution in each of the splits, both per patient and per slices. From this table we can clearly see a class imbalance, the class 1 being the minority class. To try to counteract the effect of the class imbalance, weights are employed in the loss function. The weights are calculated using the `class_weight.compute_class_weight` method from the `sklearn` library with the settings 'balanced', which is based on the paper by Foo et al. [67].

Table C.1: Detailed description of the distribution of classes per patient and slices in each of the folds.

		Patients (number)		**Patients (%)**		**Slices (number)**		**Slices (%)**	
		Train	Validation	Train	Validation	Train	Validation	Train	Validation
Fold 1	Class 0	147	37	65 %	65 %	1478	450	67%	68%
	Class 1	80	20	35 %	35 %	743	214	33%	32%
	Total	227	57			2221	664		
Fold 2	Class 0	147	37	65%	65 %	1544	384	67 %	64 %
	Class 1	80	20	35%	35 %	738	219	32 %	36 %
	Total	227	57			2282	603		
Fold 3	Class 0	147	37	65 %	65 %	1531	397	66 %	72 %
	Class 1	80	20	35 %	35 %	806	151	34 %	28 %
	Total	227	57			2337	548		
Fold 4	Class 0	147	37	65 %	65 %	1537	391	67 %	65 %
	Class 1	80	20	35 %	35 %	750	207	33 %	35 %
	Total	227	57			2287	598		
Fold 5	Class 0	148	36	65 %	65 %	1622	306	67 %	65 %
	Class 1	80	20	35 %	35 %	791	166	33 %	35 %
	Total	228	56			2413	472		

D

Assessing stopping criterion

The stopping criterion to select the best model out of the 300 trained epochs relied on selecting the model with the highest F1-score on the validation set. However, the validation set only contains 57 patients. Such as small validation set may fail in generalizing the results into an independent dataset. To observe the correlation between the validation and the independent test set, the following experiment has been done.

The CNN architecture is trained with the first cross-validation fold during 300 epochs. The model was saved every five epochs. Then, each of the saved models was loaded and used to predict the 1p/19q co-deletion status from the patients of the validation test. The same procedure was also employed to predict the 1p/19q co-deletion status of the patients of the test set. Then, the predicted probabilities were used to compute the performance metrics at each saved epoch for both validation and test set. Figures D.1 to D.6 report each of the studied performance metrics (F1-score, AUC, sensitivity, specificity, precision and accuracy) over the 300 trained models, comparing the values between the test and the validation datasets. Results suggest that there might be no direct correlation between the best classifier of such a small validation set and an independent test set. In fact, the maximum value of the f1-score of the test set does not correspond to the one of the validation set. Therefore, further research on the stopping criterion or a bigger validation set are required.

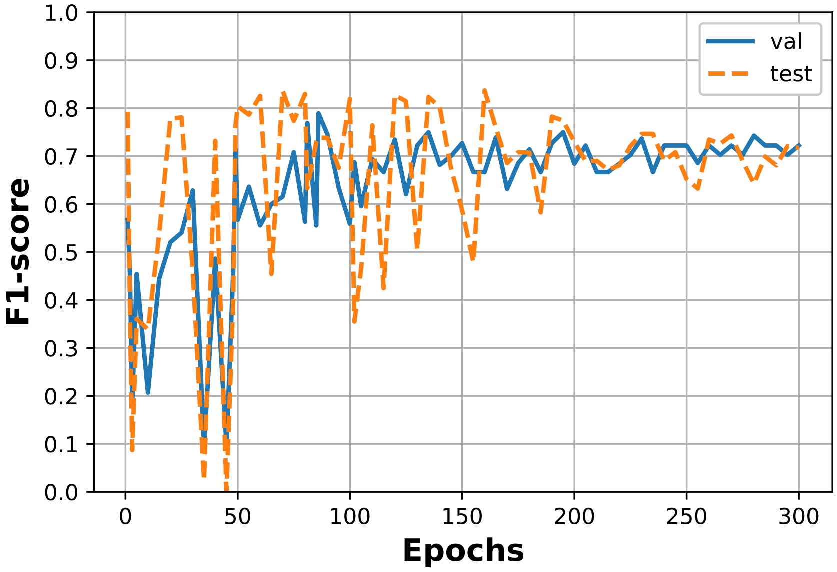


Figure D.1: Comparison of the f1-score over the 300 epochs of the validation (val) and test set.

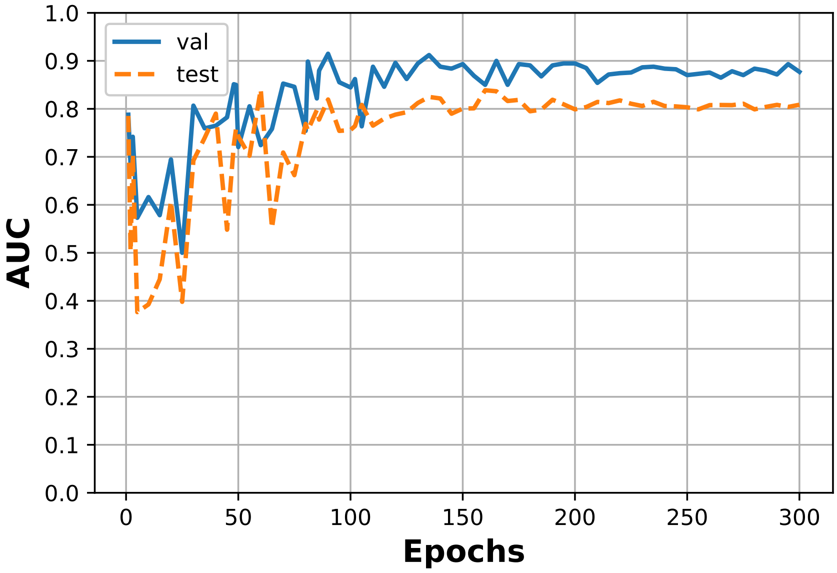


Figure D.2: Comparison of the AUC over the 300 epochs of the validation (val) and test set.

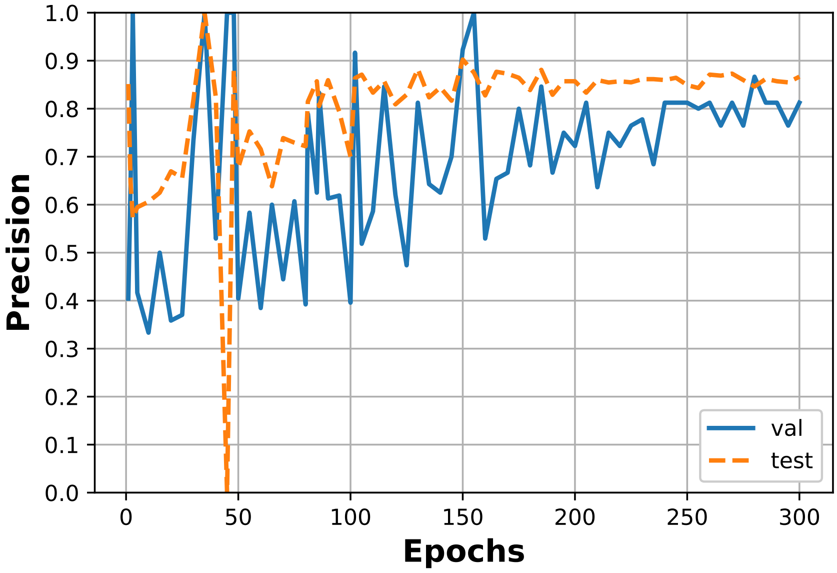


Figure D.3: Comparison of the precision over the 300 epochs of the validation (val) and test set.

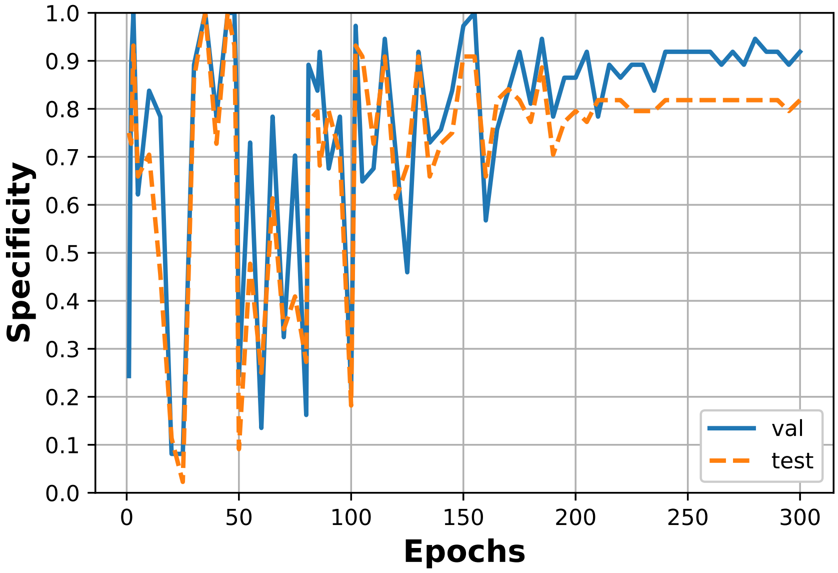


Figure D.4: Comparison of the specificity over the 300 epochs of the validation (val) and test set.

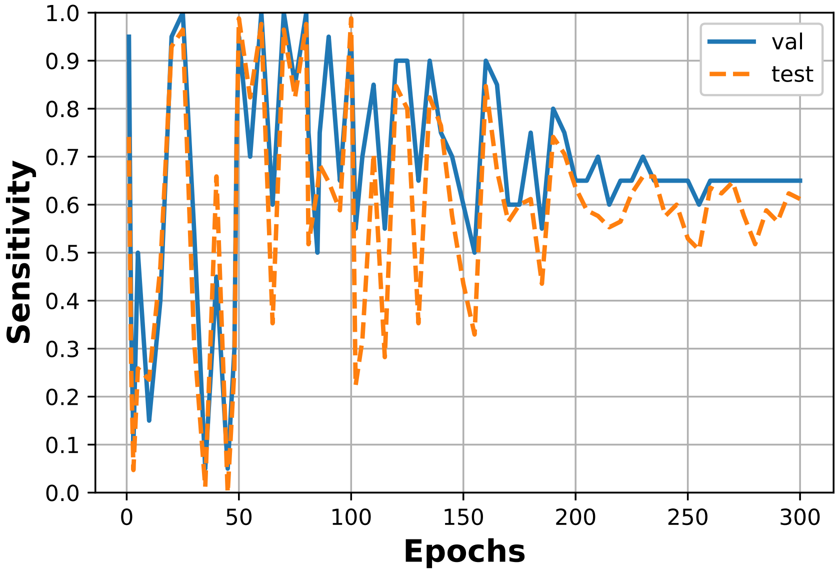


Figure D.5: Comparison of the sensitivity over the 300 epochs of the validation (val) and test set.

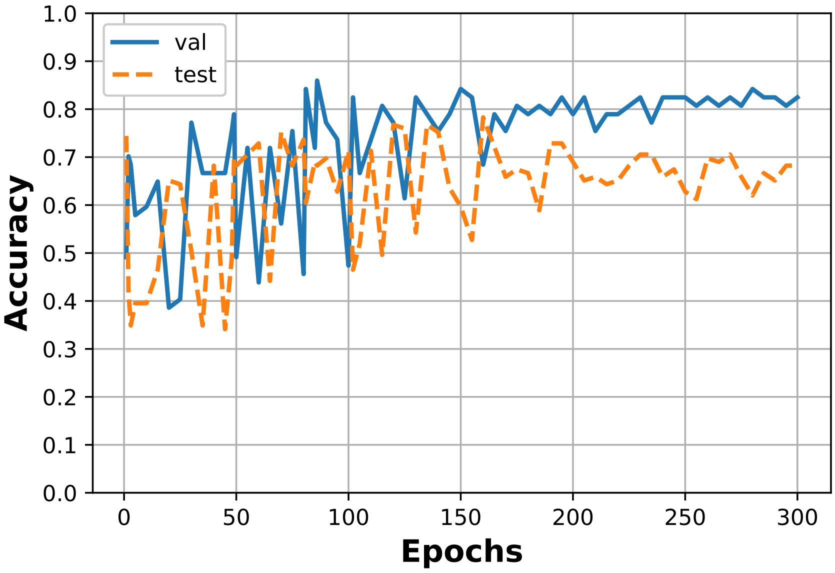


Figure D.6: Comparison of the accuracy over the 300 epochs of the validation (val) and test set.

E

Training with all the data

Table E.1: Comparison of the performance metrics of the ensemble classifier and the classifier trained with all the patients on the TCIA dataset. The CNN architecture with T2-weighted images is used.

CNN	Ensemble classifier	All patients classifier
F1-score	0.657	0.838
AUC	0.861	0.845
Precision	0.898	0.854
Specificity	0.886	0.727
Sensitivity	0.518	0.824
Accuracy	0.643	0.791

F

Stochascity of the CNN

Table F.1: General study of the stochascity of the CNN described in the report (the only difference is that L2-regularization is not employed in this case). The performance metrics of the same exact architecture is launched four different times using the fifth fold of the training EMC/HMC dataset and the T2-weighted images.

CNN	Trial 1	Trial 2	Trial 3	Trial 4	Mean	Standard Deviation
F1-score	0.879	0.911	0.876	0.897	0.891	0.016
AUC	0.861	0.861	0.889	0.917	0.882	0.027
Precision	0.750	0.737	0.789	0.824	0.774	0.039
Specificity	0.750	0.718	0.769	0.757	0.748	0.022
Sensitivity	0.750	0.700	0.750	0.700	0.725	0.029
Accuracy	0.821	0.804	0.839	0.839	0.826	0.017

Bibliography

[1] "Cancer research uk." `http://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type`. Accessed: 06-2018.

[2] E. Marieb, R. Elaine N. Marieb, and M. Katja Hoehn, *Human Anatomy & Physiology*. Benjamin-Cummings Publishing Company, 2012.

[3] D. N. Louis, H. Ohgaki, O. D. Wiestler, W. K. Cavenee, P. C. Burger, A. Jouvet, B. W. Scheithauer, and P. Kleihues, "The 2007 who classification of tumours of the central nervous system," *Acta neuropathologica*, vol. 114, no. 2, pp. 97–109, 2007.

[4] M. Smits and M. J. van den Bent, "Imaging correlates of adult glioma genotypes," *Radiology*, vol. 284, no. 2, pp. 316–331, 2017.

[5] D. A. Forst, B. V. Nahed, J. S. Loeffler, and T. T. Batchelor, "Low-grade gliomas," *The oncologist*, vol. 19, no. 4, pp. 403–413, 2014.

[6] R. Stupp, W. P. Mason, M. J. Van Den Bent, M. Weller, B. Fisher, M. J. Taphoorn, K. Belanger, A. A. Brandes, C. Marosi, U. Bogdahn, *et al.*, "Radiotherapy plus concomitant and adjuvant temozolomide for glioblastoma," *New England Journal of Medicine*, vol. 352, no. 10, pp. 987–996, 2005.

[7] T. Zeng, D. Cui, and L. Gao, "Glioma: an overview of current classifications, characteristics, molecular biology and target therapies.," *Frontiers in bioscience (Landmark edition)*, vol. 20, pp. 1104–1115, 2015.

[8] D. N. Louis, A. Perry, G. Reifenberger, A. Von Deimling, D. Figarella-Branger, W. K. Cavenee, H. Ohgaki, O. D. Wiestler, P. Kleihues, and D. W. Ellison, "The 2016 world health organization classification of tumors of the central nervous system: a summary," *Acta neuropathologica*, vol. 131, no. 6, pp. 803–820, 2016.

[9] P. L. Nguyen, D. Schultz, A. A. Renshaw, R. T. Vollmer, W. R. Welch, K. Cote, and A. V. D'Amico, "The impact of pathology review on treatment recommendations for patients with adenocarcinoma of the prostate," in *Urologic Oncology: Seminars and Original Investigations*, vol. 22, pp. 295–299, Elsevier, 2004.

[10] V. L. Staradub, K. A. Messenger, N. Hao, E. L. Wiely, and M. Morrow, "Changes in breast cancer therapy because of pathology second opinions," *Annals of Surgical Oncology*, vol. 9, no. 10, pp. 982–987, 2002.

[11] E. Manion, M. B. Cohen, and J. Weydert, "Mandatory second opinion in surgical pathology referral material: clinical consequences of major disagreements," *The American journal of surgical pathology*, vol. 32, no. 5, pp. 732–737, 2008.

[12] M. J. van den Bent, A. F. Carpentier, A. A. Brandes, M. Sanson, M. J. Taphoorn, H. J. Bernsen, M. Frenay, C. C. Tijssen, W. Grisold, L. Sipos, *et al.*, "Adjuvant procarbazine, lomustine, and vincristine improves progression-free survival but not overall survival in newly diagnosed anaplastic oligodendrogliomas and oligoastrocytomas: a randomized european organisation for research and treatment of cancer phase iii trial," *Journal of Clinical Oncology*, vol. 24, no. 18, pp. 2715–2722, 2006.

[13] G. Cairncross, B. Berkey, E. Shaw, R. Jenkins, B. Scheithauer, D. Brachman, J. Buckner, K. Fink, L. Souhami, N. Laperierre, *et al.*, "Phase iii trial of chemotherapy plus radiotherapy compared with radiotherapy alone for pure and mixed anaplastic oligodendroglioma: Intergroup radiation therapy oncology group trial 9402," *Journal of Clinical Oncology*, vol. 24, no. 18, pp. 2707–2714, 2006.

[14] A. Olar, K. M. Wani, K. D. Alfaro-Munoz, L. E. Heathcock, H. F. van Thuijl, M. R. Gilbert, T. S. Armstrong, E. P. Sulman, D. P. Cahill, E. Vera-Bolanos, *et al.*, "Idh mutation status and role of who grade and mitotic index in overall survival in grade ii–iii diffuse gliomas," *Acta neuropathologica*, vol. 129, no. 4, pp. 585–596, 2015.

[15] C. G. A. R. Network *et al.*, “Comprehensive genomic characterization defines human glioblastoma genes and core pathways,” *Nature*, vol. 455, no. 7216, p. 1061, 2008.

[16] M. Gerlinger, A. J. Rowan, S. Horswell, J. Larkin, D. Endesfelder, E. Gronroos, P. Martinez, N. Matthews, A. Stewart, P. Tarpey, *et al.*, “Intratumor heterogeneity and branched evolution revealed by multiregion sequencing,” *New England journal of medicine*, vol. 366, no. 10, pp. 883–892, 2012.

[17] D. W. Parsons, S. Jones, X. Zhang, J. C.-H. Lin, R. J. Leary, P. Angenendt, P. Mankoo, H. Carter, I.-M. Siu, G. L. Gallia, *et al.*, “An integrated genomic analysis of human glioblastoma multiforme,” *Science*, vol. 321, no. 5897, pp. 1807–1812, 2008.

[18] T. Sjöblom, S. Jones, L. D. Wood, D. W. Parsons, J. Lin, T. D. Barber, D. Mandelker, R. J. Leary, J. Ptak, N. Silliman, *et al.*, “The consensus coding sequences of human breast and colorectal cancers,” *science*, vol. 314, no. 5797, pp. 268–274, 2006.

[19] N. Navin, J. Kendall, J. Troge, P. Andrews, L. Rodgers, J. McIndoo, K. Cook, A. Stepansky, D. Levy, D. Esposito, *et al.*, “Tumour evolution inferred by single-cell sequencing,” *Nature*, vol. 472, no. 7341, p. 90, 2011.

[20] “The cost of sequencing a human genome, national human genome research institute (nhgri).” `https://www.genome.gov/27565109/the-cost-of-sequencing-a-human-genome/`. Accessed: 03-2018.

[21] E. W. Amundson, M. J. McGirt, and A. Olivi, “A contralateral, transfrontal, extraventricular approach to stereotactic brainstem biopsy procedures,” *Journal of neurosurgery*, vol. 102, no. 3, pp. 565–570, 2005.

[22] G. Shukla, G. S. Alexander, S. Bakas, R. Nikam, K. Talekar, J. D. Palmer, and W. Shi, “Advanced magnetic resonance imaging in glioblastoma: a review,” *Chinese clinical oncology*, vol. 6, no. 4, 2017.

[23] “Visually accessable rembrandt [repository for molecular brain neoplasia data] images.” `https://wiki.cancerimagingarchive.net/display/Public/VASARI+Research+Project`. Accessed: 03-2018.

[24] H. J. Aerts, E. R. Velazquez, R. T. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, *et al.*, “Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach,” *Nature communications*, vol. 5, p. 4006, 2014.

[25] E. Sala, E. Mema, Y. Himoto, H. Veeraraghavan, J. Brenton, A. Snyder, B. Weigelt, and H. Vargas, “Unravelling tumour heterogeneity using next-generation imaging: radiomics, radiogenomics, and habitat imaging,” *Clinical radiology*, vol. 72, no. 1, pp. 3–10, 2017.

[26] R. J. Gillies, P. E. Kinahan, and H. Hricak, “Radiomics: images are more than pictures, they are data,” *Radiology*, vol. 278, no. 2, pp. 563–577, 2015.

[27] M. Avanzo, J. Stancanello, and I. El Naqa, “Beyond imaging: The promise of radiomics,” *Physica Medica: European Journal of Medical Physics*, vol. 38, pp. 122–139, 2017.

[28] R. T. Larue, G. Defraene, D. De Ruysscher, P. Lambin, and W. Van Elmpt, “Quantitative radiomics studies for tissue characterization: a review of technology and methodological procedures,” *The British journal of radiology*, vol. 90, no. 1070, p. 20160665, 2017.

[29] O. Gevaert, J. Xu, C. D. Hoang, A. N. Leung, Y. Xu, A. Quon, D. L. Rubin, S. Napel, and S. K. Plevritis, “Non–small cell lung cancer: identifying prognostic imaging biomarkers by leveraging public gene expression microarray data—methods and preliminary results,” *Radiology*, vol. 264, no. 2, pp. 387–396, 2012.

[30] M. Diehn, C. Nardini, D. S. Wang, S. McGovern, M. Jayaraman, Y. Liang, K. Aldape, S. Cha, and M. D. Kuo, “Identification of noninvasive imaging surrogates for brain tumor gene-expression modules,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 13, pp. 5213–5218, 2008.

[31] R. W. Jansen, P. van Amstel, R. M. Martens, I. E. Kooi, P. Wesseling, A. J. de Langen, C. W. Menke-Van der Houven, *et al.*, “Non-invasive tumor genotyping using radiogenomic biomarkers, a systematic review and oncology-wide pathway analysis,” *Oncotarget*, vol. 9, no. 28, p. 20134, 2018.

[32] S. R. van der Voort, R. Gahrmann, M. J. van den Bent, A. J. Vincent, W. J. Niessen, M. Smits, and S. Klein, “Radiogenomic classification of the 1p/19q status in presumed low-grade gliomas,” in *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pp. 638–641, IEEE, 2017.

[33] B. Shofty, M. Artzi, D. B. Bashat, G. Liberman, O. Haim, A. Kashanian, F. Bokstein, D. T. Blumenthal, Z. Ram, and T. Shahar, “Mri radiomics analysis of molecular alterations in low-grade gliomas,” *International journal of computer assisted radiology and surgery*, pp. 1–9, 2017.

[34] H. Zhou, M. Vallières, H. X. Bai, C. Su, H. Tang, D. Oldridge, Z. Zhang, B. Xiao, W. Liao, Y. Tao, *et al.*, “Mri features predict survival and molecular markers in diffuse lower-grade gliomas,” *Neuro-oncology*, vol. 19, no. 6, pp. 862–870, 2017.

[35] C.-F. Lu, F.-T. Hsu, K. L.-C. Hsieh, Y.-C. J. Kao, S.-J. Cheng, J. B.-K. Hsu, P.-H. Tsai, R.-J. Chen, C.-C. Huang, Y. Yen, *et al.*, “Machine learning-based radiomics for molecular subtyping of gliomas,” *Clinical Cancer Research*, pp. clincanres–3445, 2018.

[36] Y. LeCun, C. Cortes, and C. J. C. Burges, “MNIST repository.” `http://yann.lecun.com/exdb/mnist/`. Accessed: 12-2018.

[37] Z. Akkus, I. Ali, J. Sedlar, T. L. Kline, J. P. Agrawal, I. F. Parney, C. Giannini, and B. J. Erickson, “Predicting 1p19q chromosomal deletion of low-grade gliomas from mr images using deep learning,” *arXiv preprint arXiv:1611.06939*, 2016.

[38] K. Chang, H. X. Bai, H. Zhou, C. Su, W. L. Bi, E. Agbodza, V. K. Kavouridis, J. T. Senders, A. Boaro, A. Beers, *et al.*, “Residual convolutional neural network for the determination of idh status in low-and high-grade gliomas from mr imaging,” *Clinical Cancer Research*, vol. 24, no. 5, pp. 1073–1081, 2018.

[39] R. A. Pooley, “Fundamental physics of mr imaging,” *Radiographics*, vol. 25, no. 4, pp. 1087–1099, 2005.

[40] S. Abrol, A. Kotrotsou, A. Salem, P. O. Zinn, and R. R. Colen, “Radiomic phenotyping in brain cancer to unravel hidden information in medical images,” *Topics in Magnetic Resonance Imaging*, vol. 26, no. 1, pp. 43–53, 2017.

[41] M. Incoronato, M. Aiello, T. Infante, C. Cavaliere, A. M. Grimaldi, P. Mirabelli, S. Monti, and M. Salvatore, “Radiogenomic analysis of oncological data: a technical survey,” *International journal of molecular sciences*, vol. 18, no. 4, p. 805, 2017.

[42] E. R. Velazquez, H. J. Aerts, Y. Gu, D. B. Goldgof, D. De Ruysscher, A. Dekker, R. Korn, R. J. Gillies, and P. Lambin, “A semiautomatic ct-based ensemble segmentation of lung tumors: Comparison with oncologists’ delineations and with the surgical specimen,” *Radiotherapy and Oncology*, vol. 105, no. 2, pp. 167–173, 2012.

[43] P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview,” *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.

[44] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (roc) curve.,” *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[45] T. J. Sejnowski and C. R. Rosenberg, “Parallel networks that learn to pronounce english text,” *Complex systems*, vol. 1, no. 1, pp. 145–168, 1987.

[46] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.

[47] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[48] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[49] Y. LeCun, Y. Bengio, *et al.*, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[50] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[51] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[52] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[53] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[54] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, “Local binary convolutional neural networks,” in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, vol. 1, IEEE, 2017.

[55] C. O’Connor, “Fluorescence in situ hybridization (fish),” 2008. [Online; accessed 2018-08-13].

[56] J. S. Reis-Filho, “Next-generation sequencing,” *Breast Cancer Research*, vol. 11, no. 3, p. S12, 2009.

[57] J. P. Pluim, J. A. Maintz, and M. A. Viergever, “Mutual-information-based registration of medical images: a survey,” *IEEE transactions on medical imaging*, vol. 22, no. 8, pp. 986–1004, 2003.

[58] K. Marstal, F. Berendsen, M. Staring, and S. Klein, “Simpleelastix: A user-friendly, multi-lingual library for medical image registration,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 134–142, 2016.

[59] Z. S. J. K. P. Erickson, Bradley; Akkus, “Data from lgg-1p19qdeletion. the cancer imaging archive,” 2017. Accessed: 2018-05-01.

[60] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.

[61] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[62] P. Chang, J. Grinband, B. Weinberg, M. Bardis, M. Khy, G. Cadena, M.-Y. Su, S. Cha, C. Filippi, D. Bota, *et al.*, “Deep-learning convolutional neural networks accurately classify genetic mutations in gliomas,” *American Journal of Neuroradiology*, 2018.

[63] C. Nadeau and Y. Bengio, “Inference for the generalization error,” in *Advances in neural information processing systems*, pp. 307–313, 2000.

[64] M. Pavic, M. Bogowicz, X. Würms, S. Glatz, T. Finazzi, O. Riesterer, J. Roesch, L. Rudofsky, M. Friess, P. Veit-Haibach, *et al.*, “Influence of inter-observer delineation variability on radiomics stability in different tumor sites,” *Acta Oncologica*, pp. 1–5, 2018.

[65] S. Fellah, D. Caudal, A. De Paula, P. Dory-Lautrec, D. Figarella-Branger, O. Chinot, P. Metellus, P. Cozzone, S. Confort-Gouny, B. Ghattas, *et al.*, “Multimodal mr imaging (diffusion, perfusion, and spectroscopy): is it possible to distinguish oligodendroglial tumor grade and 1p/19q codeletion in the pretherapeutic diagnosis?,” *American Journal of Neuroradiology*, vol. 34, no. 7, pp. 1326–1333, 2013.

[66] R. Brown, M. Zlatescu, A. Sijben, G. Roldan, J. Easaw, P. Forsyth, I. Parney, R. Sevick, E. Yan, D. Demetrick, *et al.*, “The use of magnetic resonance imaging to noninvasively detect genetic signatures in oligodendroglioma,” *Clinical Cancer Research*, vol. 14, no. 8, pp. 2357–2362, 2008.

[67] G. King and L. Zeng, “Logistic regression in rare events data,” *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.