Radiomics vs. Deep Learning to predict lipomatous soft tissue tumors malignancy on Magnetic Resonance Imaging

---

Guillaume Fradet

Master 2, Data Science

Altran Research, Vélizy-Villacoublay

EILiS department

Academic tutors: Erwan Le Pennec, Mohamad Ghassany

Company tutor: Reina Ayde

September 2019 - Non-confidential document

Acknowledgements

I would like to express my deepest gratitude to my company tutor, Reina Ayde, who guided me in this great research project, that would not have been possible without her. Many thanks also for our talks about my future, that helped me to identify clearly what I was looking for. I also thank Hugo Bottois, who helped me to review this report.

I would like to acknowledge Benjamin Leporq, our CREATIS interlocutor, for our collaboration, our exchanges, and for his time spent on the preparation of the datasets, such as the tumor segmentation.

Many thanks to all the interns for what we shared together, lots of fun but also precious knowledge about deep learning.

A special appreciation to Mohamad Ghassany, my ESILV's tutor, and first professor to teach me machine learning. He made me discover this field in such a way that it quickly became a vocation.

Last but not least, my thanks go to Erwan Le Pennec, head of the Master 2 in Data Science, for the quality of the formation and the opportunities that it opened.

Déclaration d'intégrité relative au plagiat

Je soussigné Guillaume FRADET certifie sur l'honneur :

1. Que les résultats décrits dans ce rapport sont l'aboutissement de mon travail.
2. Que je suis l'auteur de ce rapport.
3. Que je n'ai pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les règles bibliographiques préconisées.

Je déclare que ce travail ne peut être suspecté de plagiat.

Date : 18/09/2019

Signature : Fradet.

Contents

Abstract

Introduction: Lipomatous soft-tissue tumors grow from mesenchymal tissue, referred as lipoma and liposarcoma for benign and malignant tumors respectively. Five subclasses of liposarcoma exist, requiring different patient treatments. Most of the types are easily distinguishable relying on Magnetic Resonance Imaging (MRI). But lipoma and Atypical Lipomatous Tumor / Well-Differentiated Liposarcoma (ALT/WDL) have overlapping MR imaging characteristics. A biopsy is usually performed to detect cancerous cells, and classify the tumor. However, these biopsies are invasive, costly and unnecessary in most cases as the malignant/benign ratio is significantly low (1/100). This work aim to provide efficient MRI-based decision support tools to discriminate cancerous tumors, as an alternative to biopsies.

Materials et methods: 85 MRI scans from patients with lipoma or ALT/WDL were gathered from 43 different centers with non-uniform protocols. We compared different approaches based on three versions of the MRI dataset: a 2D version with only one slice per patient (where the tumor is the largest), a 3D version (with all the slices where the tumor is visible), and a 3D version with batch-effect correction, to remove the inter-site technical variability. Radiomic features were extracted from the datasets, producing respectively 35 and 92 features for the 2D and 3D collection. In parallel, the MR images were normalized for pixel intensity, and inhomogeneities were corrected. We compared traditional machine learning algorithms based on radiomic data, to deep learning approaches using Convolutional Neural Networks (CNN) applied directly on MR images. Three CNN-based architectures were compared: a custom CNN learned from scratch, a fine-tuned pre-trained ResNet50 model, and a XGBoost classifier based on a CNN feature extraction.

Results: The models performance were assessed using 10 cross-validation folds. On the batch-corrected 3D radiomic dataset, we achieve to classify correctly all the validation folds (mean AUROC = 1) using linear Support Vector Machine (SVM) with feature selection, as well as using a Random Forest classifier. The best image classification performance was obtained by fine-tuning the pre-trained ResNet50 (mean AUROC = 0.878, std AUROC = 0.11).

Conclusion: In our context of very limited observations, radiomic-based models outperformed the image-based approaches. Importantly, the batch-effect normalization, that removed the inter-site technical variability on the radiomic data, had a huge effect on the models performance. With such a small dataset, it was a hard task to train complex architectures like CNN. Moreover, the MRI scans were acquired on various body regions, resulting in high heterogeneity in the images, making the generalization even harder. These exciting results on the radiomics need to be confirmed on an external validation cohort, but could have an impact on clinical practice to differentiate lipoma from ALT/WDL based only on MRI, and in a wider approach, to classify all types of lipomatous soft-tissue tumors.

Introduction

This six months internship took place in Altran Research. The latter was created in 2009 by the global engineering consulting firm Altran, in order to boost their ability to innovate for their clients. I joined the EILiS division, which stands for Energy, Industry and Life Sciences, to work on a e-health project.

This project was focusing on lipomatous soft-tissue tumors. These tumors grow from mesenchymal tissue and can occur anywhere in the body. They commonly occur in the neck, shoulders, back, abdomen, arms and thighs. A lipoma is a benign lipomatous tumor and the most common one. It is not a cancer and is usually harmless. A liposarcoma is a malignant lipomatous tumor. It is a rare type of cancer that begins in the fat cells[1]. Liposarcoma accounts for approximately 20% of sarcoma in adults; therefore, it is the most frequently encountered malignant soft-tissue tumor in clinical practice [1]. In 2002, the World Health Organization published a classification of soft-tissue tumors, subdividing liposarcoma into five classes: well-differentiated, dedifferentiated, myxoid, pleomorphic, and mixed liposarcomas. Atypical lipomatous tumor or well differentiated liposarcoma (ALT/WDL) is the most common liposarcoma among the subclasses. It represents 40 to 45% of liposarcomas [2].

The patient treatment differs depending on the type of the tumor. Most of the types are easily distinguishable relying on Magnetic Resonance Imaging (MRI). But ALT/WDL and lipoma have overlapping MR imaging characteristics, meaning that their MRI appearances is highly similar [3]. It is essential to differentiate these two tumor types, because a lipoma can be treated with a marginal excision or even simply by surveillance (if it doesn't provide any discomfort to the patient), but an ALT/WDL must be removed by complete resection.

[1] Also known as adipocytes or lipocytes, these cells are specialized in the storage of fat.

Currently, the strategy to distinguish ALT/WDL from lipomas is based on biopsy, followed by histological study. However, this practice is costly and invasive for the patient. Moreover, the majority of these tumors are benign. In fact, benign mesenchymal tumours outnumber sarcomas by a factor of at least 100 [2]. Therefore, most of these biopsies are unnecessary.

In recent years, a field of medical imaging analysis, called *radiomics*, has emerged. The latter consists in translating medical images into complex and high-dimensional quantitative data, using data-characterization algorithms [4]. The obtained features are then analysed by machine learning techniques.

On the other hand, convolutional deep learning methods, like convolutional neural network (CNN), are applied directly on images. These methods have shown excellent performance, and have already beaten the human-level performance on tasks like image classification on large-scaled dataset [5]. The idea to apply such methods in the context of medical imaging seems appealing. However, huge amount of data is usually necessary to train such networks. Yet, it is a challenge to build large datasets in the medical context, mainly because the annotation of medical imaging requires expert knowledge from radiologists.

First, in the hope to find an alternative to costly and invasive biopsies, this project aims to find the best MRI-based machine learning strategy to classify the malignancy of lipomatous soft-tissue tumors. Secondly, we wish to answer a more general question in a context where the amount of samples available for the training and the validation of models is very limited: which approach performs the best between a classification based on radiomic data, and deep learning methods applied directly on MR images.

A collaboration with the biomedical imaging research laboratory CREATIS *(Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé)* made this project possible. In particular, they gathered the clinical data, and extracted the radiomic features from the medical images [6], so that we could focus on the machine learning part.

This report is split into three sections. The first section introduces the dataset, giving insights about the two forms: radiomics and MR images. The second section focuses on the task of classifying the tumor malignancy based on radiomics, comparing different machine learning methods. Finally, the last part covers deep learning methods applied on the MR images.

The datasets

Our study was based on a private MRI dataset, built by the research laboratory CREATIS. 85 patients with lipomatous soft tissue tumors formed the dataset. The tumors were either lipomas or atypical lipomatous tumors / well-differentiated liposarcomas (ALT/WDL). They were labeled by histology. The classes were approximately balanced, as the collection contained 40 lipomas and 45 ALT/WDL. The MR images were acquired with T1-weighted sequences, and gadolinium-contrast enhanced.

A MRI scan produces multiple 2D slices. During the majority of the internship, we had access to only one slice for each patient: the one where the tumor was the largest. In this report, we will refer to this version as the 2D dataset. Later, we obtained all the slices were the tumor was visible. We call this collection the 3D dataset.

It is essential to note the important heterogeneity in our collection. First, as these types of tumor can occur anywhere in the body, the slices were acquired on various body regions. Secondly, the gathered data came from 43 different centers with non-uniform protocols, 16 different MR systems from four manufacturers – General Electric (38.8%), Siemens (37.6%), Philips (21.2%), Toshiba (2.4%) – and using three different static fields (1.0T, 1.5T, 3.0T).

This dataset was under two forms: as radiomic features, or as raw MR images. In the next sections, we present the latter.

2.1 Radiomics

Radiomics is an emerging field of medical imaging analysis, applicable to tomographic images, i.e. computed tomography (CT), magnetic resonance (MR), or positron emission tomography (PET) images. The idea is to transform a raw clinical image (that

can either be two-dimensional or three-dimensional) into a set of high-dimensional mineable features which characterize this image. One could then build a model for decision support, based on these extracted features, or even combined with additional data like genomics, patient's history, clinical features, etc. Indeed, multimodal predictive modeling could enhance a personalized medicine, that would ensure better patient care.

The process of radiomics is composed of multiple discrete steps: *(a)* acquiring the images, *(b)* segmenting the area / volume of interest (i.e. delineating the borders of the tumor), *(c)* extracting the radiomic features, *(d)* mining these data to develop classifier models. Parts *(b)* and *(c)* were executed by CREATIS laboratory. Images were automatically loaded in in-house software developed on Matlab R2019a (The MathWorks, Natick, USA). The tumor was manually segmented in three dimension, slice-by-slice, by an experimented physicist (with 15 years experiences in MR imaging). Tumor mask was next applied on fat-suppressed enhanced MR image and 92 quantitative radiomic features were extracted. The feature extraction was performed by automated algorithms, and resulted in two groups of features: *semantic* and *agnostic*. Semantic features are the common statistics to describe a tumor, e.g. by its size and shape. Agnostic features are less common for radiologists. They are first-, second- and higher-order statistics, that respectively describe the distribution of values of individual voxels (based on histogram methods), the statistical interrelationships between voxels with similar (or dissimilar) contrast values, the repetitive or nonrepetitive patterns [7]. The full list of features is summarized in Figure 2.1.

Size and shape features were directly extracted from the binary mask. Intensity distribution features were extracted from masked MR images without normalization or filtering of voxel intensities and from the histogram built with 256 bins. Before the extraction of texture features, images gray levels were discretized in a smaller number of gray levels. This operation was done using an equal probability algorithm to define decision thresholds in the volume such as the number of voxels for a given reconstructed level is the same in the quantized volume for all gray levels. Images were discretized in 8, 16, 24, 32, 40, 48 and 64 grey levels and for each level four matrix were built: GLCM (Gray-level co-occurrence matrix), GLRLM (Gray-level run length matrix), GLSZM (Gray-level size zone matrix) and GLSZM (Neighborhood gray tone difference matrix) from which characteristics were extracted according to [8–14]. Frequency domain-based texture features were extracted using a Gabor filtering. GLCM and GLRLM were computed for 4 directions (0°, 45°, 90° and 135°)

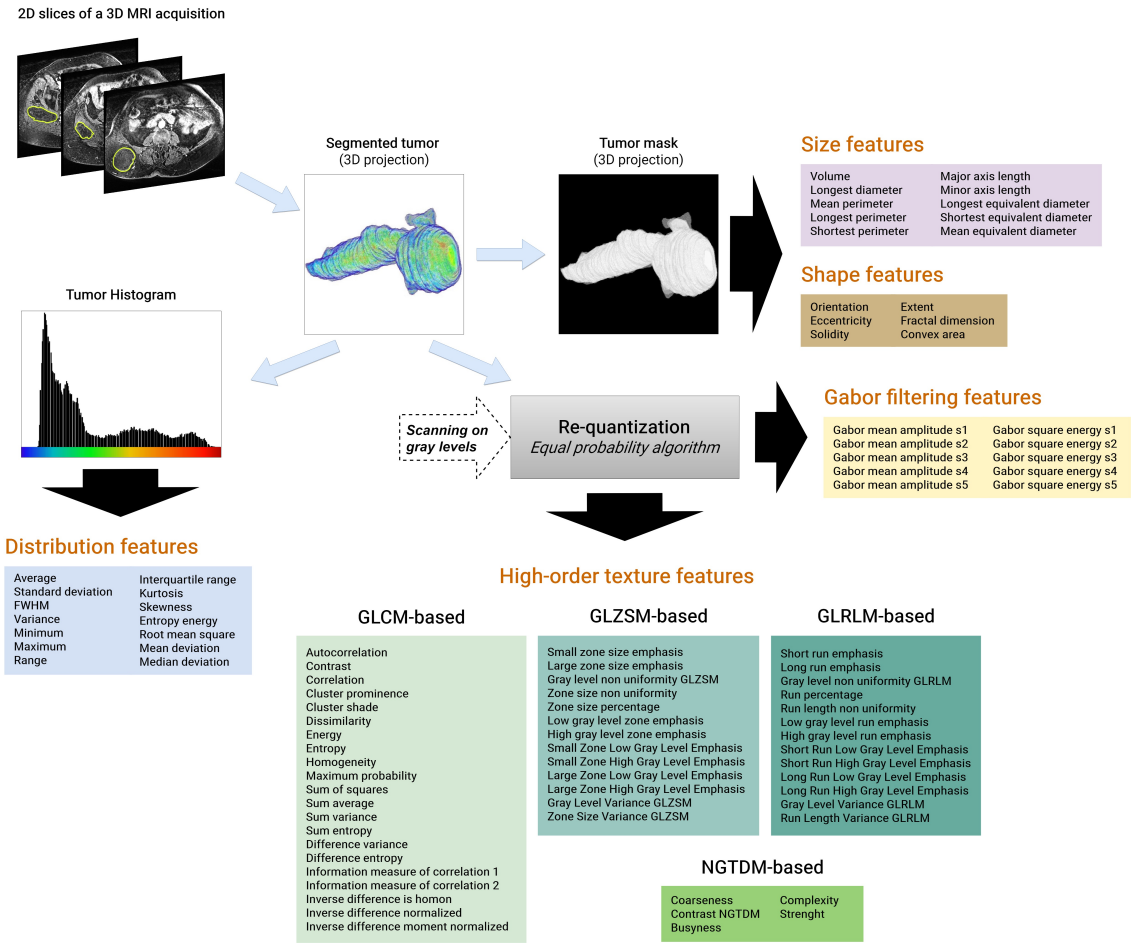


Figure 2.1: Radiomic extraction pipeline.

with an offset of 1 pixel. For GLSZM and GLSZM, a 26 pixel connectivity was used. For Gabor filtering, 5 scales, 6 orientations, and a minimal wavelength of 3 were used (Fig 2.1).

Because the clinical images were acquired on multi-site with different MR systems, an important part was to apply a post-processing data harmonization. ComBat algorithm is a popular batch-effect correction tool, that has been shown to successfully remove inter-site technical variability while preserving inter-site biological variability [15–17]. It was applied on the radiomic features, using the Matlab implementation available on GitHub[1].

During the internship, this radiomic dataset has evolved and resulted in different versions:

[1]https://github.com/Jfortin1/ComBatHarmonization

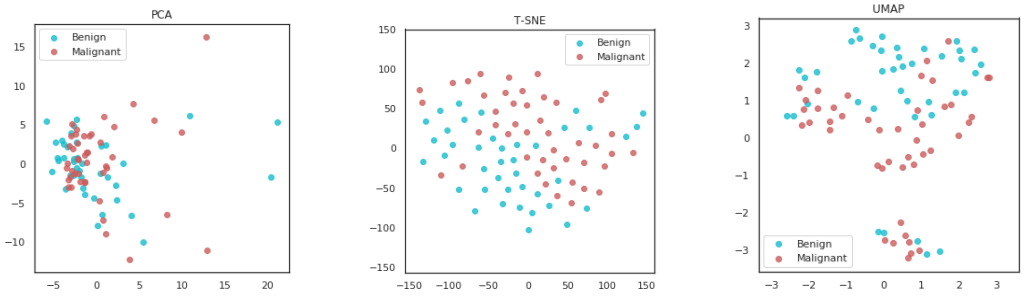
1. 2D radiomics, applied on the slice where the tumor area is the largest, without batch-effect normalization (81 patients, 35 features)
2. 3D radiomics, applied on all slices where the tumor is visible, without batch-effect normalization (85 patients, 92 features)
3. 3D radiomics, applied on all slices where the tumor is visible, with batch-effect normalization (85 patients, 92 features)

Naturally, some size features are altered between the 2D and 3D radiomics. The tumor area in the 2D dataset becomes the tumor volume. From the single perimeter and equivalent diameter in 2D, the mean, the longest, and the shortest one are kept in 3D. This results in a more authentic representation of the tumor, and therefore better characteristics on which one could build a trustworthy decision support model.

Exploratory data analysis

The target variable y was encoded as ordinal integers: 0 for a lipoma (benign tumor), 1 for an ALT/WDL (malignant tumor). The tumor size features were, in average, well correlated with the target (e.g. Pearson correlation coefficient of 0.49 between y and *longest perimeter*). In the high-order texture features, the *Large zone low gray level emphasis* stands out, with a correlation coefficient of 0.63, that was the most correlated feature with the target. In the subgroups of features (Fig 2.1), many attributes were highly correlated between each other. See Figures A.1, A.2 in the appendix (pages 31-32).

Using non-supervised dimensionality reduction methods (Fig 2.2) like t-SNE [18] or UMAP [19] for visualization, we start to distinguish the two classes.

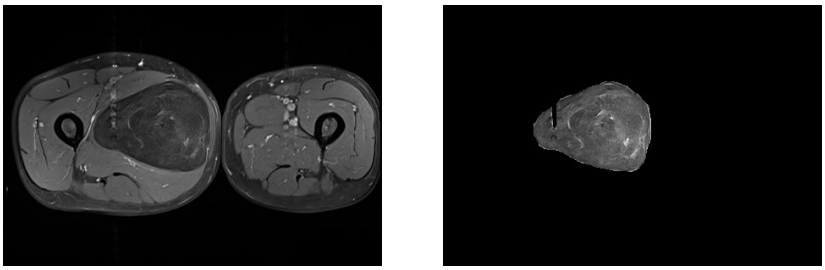


(a) PCA (b) t-SNE (c) UMAP

Figure 2.2: 2D projection of dimensionality reduction applied on 3D radiomics with batch-effect normalization (after standardization).

2.2 Magnetic resonance imaging

On each MRI slice, a tumor segmentation was performed, resulting in two computer files: the `reference_slice`, where we can see the whole slice, and the `tumor_segmentation`, where only the tumor appears. This last file is also called a *mask*.

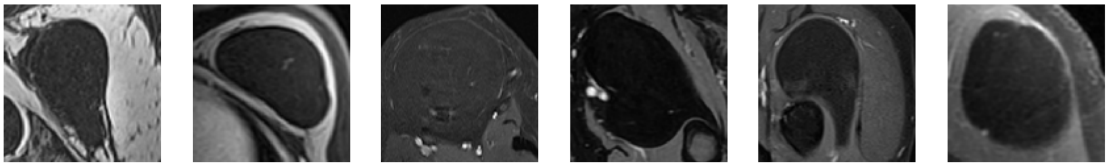


(a) `reference_slice` (b) `tumor_segmentation`

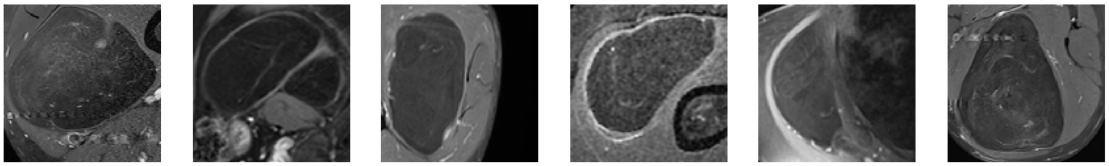
Figure 2.3: 2D slice of a malignant tumor located in the thigh.

See Figure A.3 in the appendix (page 33), for a visualization of all the MRI scans with their tumor segmentation.

As mentioned in the introduction, it is a hard task to distinguish a lipoma (benign tumor) from an ALT/WDL (malignant tumor), based on magnetic resonance imaging. On the Figure 2.4, we observe a pixel intensity variation between the various tumors. It might be caused by the MRI itself, due to the magnetic field inhomogeneities and scanner-related intensity artifacts [20].



(a) Lipomas



(b) ALT/WDL

Figure 2.4: MRI comparison between lipomas and ALT/WDL.

Moving from the 2D dataset (only one slice per patient) to the 3D dataset, we obtained data from new patients, increasing the number of patients from 81 to 87. More importantly, we had access to all the 2D slices where the tumor was visible. It augmented the dataset from 81 slices to 2721.

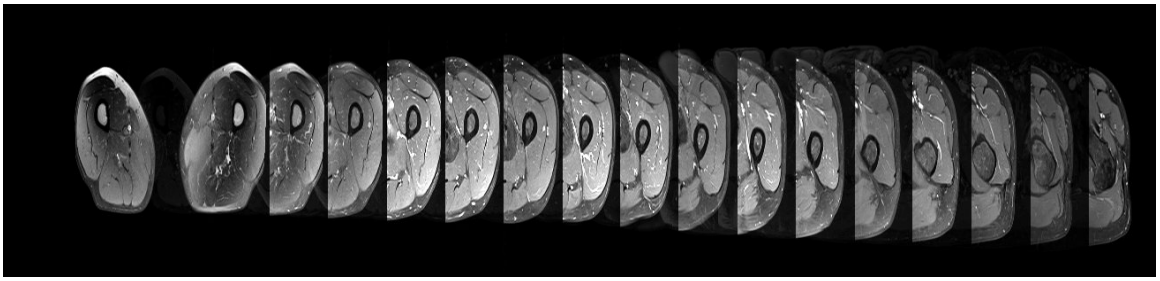


Figure 2.5: 3D projection of an MRI scan in the thighs.

Most scans had between 16 and 31 slices, which are respectively the first and third quartiles. However, some scans had more than 150 slices, the maximum being 232 slices. Usually, the malignant soft-tissue tumors are larger in volume than the benign ones. Therefore, because our 3D dataset was composed of slices where the tumors are visible, we had much more slices of malignant tumors. In fact, for one benign tumor slice, we had approximately two malignant tumor slices (893 lipomas for 1828 ALT/WDL). This dataset was imbalanced, as the malignant class was much more represented, unlike the 2D dataset with only one slice per patient.

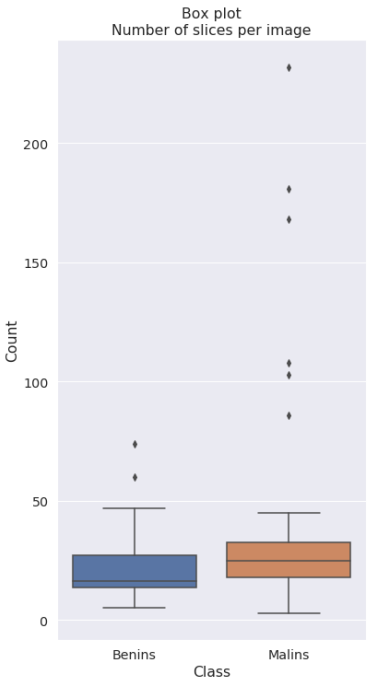


Figure 2.6: Box plot of the slices frequency per MRI scan.

Tumor classification based on Radiomics

Based on the radiomics, we compared the performance of different machine learning classifiers. The supervised task was to label the tumors, that are either a lipoma or an ALT/WDL. In the following sections, we briefly describe three classifiers – Support Vector Machine, Random Forest, and Multilayer perceptron – that were optimized for our task. Then, we compare and interpret the obtained results.

3.1 Support Vector Machine classifier

A Support Vector Machine (SVM) is a discriminative model that separates two classes with a hyperplane. The idea is to find the optimal hyperplane that best separates the classes. In other words, the goal is to maximize the margin around the separating hyperplane. It introduces kernel functions to extend to non-linearly separable patterns. The *kernel trick* consists in mapping original non-linear observations into a higher-dimensional space in which they become separable, so that we can calculate the hyperplane. To avoid overfitting, regularization can be applied on this model, by tuning a parameter called C, resulting in larger or smaller widths for the hyperplane margin. [21]

We used two *scikit-learn* [22] implementations of SVM. First, `sklearn.svm.SVC` based on *libsvm* software, giving access to multiple kernels. Then, `sklearn.svm.LinearSVC` implemented in terms of *liblinear*, similar to `SVC` with parameter `kernel='linear'`, but providing more flexibility in the choice of penalties and loss functions.
A standardization was applied on the features, by removing the mean and scaling to unit variance, so that the attributes with large numeric ranges do not dominate

attributes with smaller numeric ranges. The standardization scaler was learned only on the training set and then applied on both sets (training and validation).
Recursive feature elimination (RFE) and cross-validated selection of the best number of features were applied to find the best set of attributes. Another strategy was to apply dimensionality reduction techniques to transform the original feature space into a lower dimension. We used principal component analysis (PCA) procedure. The number of components was selected such that the amount of variance that needed to be explained was greater than a percentage (e.g. 99%).
Best SVM hyperparameters were found using grid search, iterating through different kernels (linear, radial basis function (RBF), polynomial, sigmoid) and different values for the penalty parameter C. Moreover, in the case of RBF, polymial and sigmoid kernels, the kernel coefficient γ was fine-tuned, as well as the degree of the polynomial kernel function. Finally, for the linear kernel, different loss functions (hinge, squared hinge) and penalties (L1, L2) were compared, solving the dual or primal optimization problem, using or not the intercept in the calculation.

3.2 Random Forest classifier

Random Forests [23] are based on multiple individual decision trees, learned independently. Combined, these trees construct a powerful ensemble. The forest's prediction takes into account each tree vote, and retain the most picked class.

We implemented our random forest classifier with the Python library scikit-learn [22]. No standardization was applied on the features, as it has no effect on decision trees. The number of trees in the forest was chosen with grid search, as well as the metric to measure the quality of a split in the decision trees: either Gini impurity or information gain. No maximum depth of the tree was specified such that all leaves were pure.

3.3 Multilayer Perceptron classifier

A multilayer perceptron (MLP) is an artificial neural network (ANN). It belongs to the class of feedforward networks, meaning that the information always goes forward, unlike in recurrent neural networks (RNN). MLP is based on the accumulation of multiple layers of computational units (neurons). As input, these units take the output from the previous layer neurons, or, in the case of the first layer, the values of the

features in X. Each neuron computes an output by applying an activation function on the pre-activation $z(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b$, where $\boldsymbol{x}$, $\boldsymbol{w}$ and b are respectively the input vector, the weights vector, and the bias. The neurons weights and biases are learned and adjusted through back-propagation [24]. In our case of binary classification, the final layer is a single neuron, with a sigmoid activation function: $\sigma(z) = \frac{1}{1+e^{-z}}$. This function outputs a decimal between 0 and 1, that is the probability for the sample to belong to the malignant class. Therefore, if this probability is lower than 0.5, we predict that the tumor is a lipoma, otherwise we predict an ALT/WDL.

We used the Keras [25] API, with TensorFlow [26] backend, to implement the network. A feature standardization, learned on the training set, was applied on both sets. The fully-connected layers were defined as `Dense` layers, specifying the number of neurons and an optional regularization (weight decay) to avoid overfitting. The weights were initialized with the Glorot uniform initializer [27], and the biases with zeros. A batch normalization [28] was optionally applied before the activation function, to reduce the internal covariate shift by normalizing the layer inputs. This procedure might accelerate the training because it allows us to use higher learning rates, but also to be less careful about initialization. The Rectified Linear Unit (ReLU) was the activation function for the hidden layers: $relu(z) = \max(0, z)$. A Dropout [29] could be added at the end of the block (i.e. before a new fully-connected layer). The last layer was composed of a single neuron, activated by the sigmoid function.

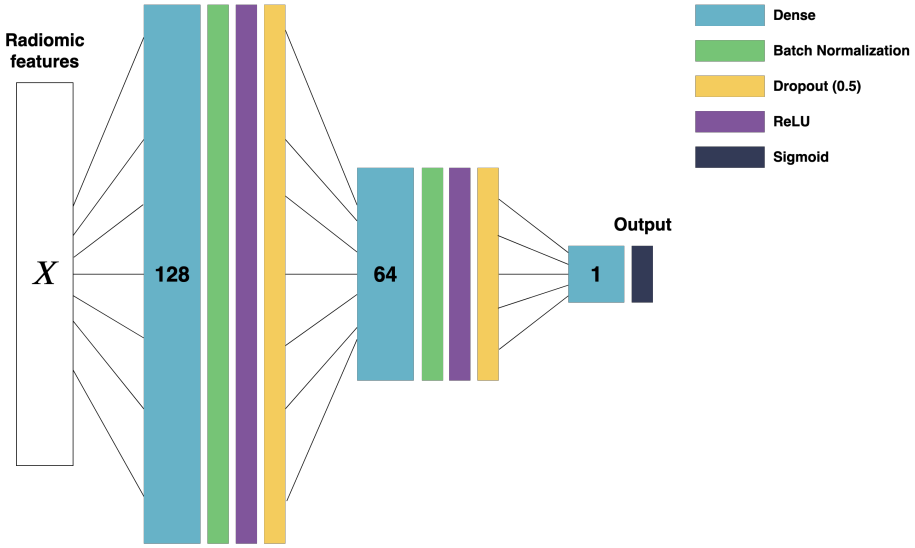


Figure 3.1: Example of a MLP architecture with two hidden layers of 128 and 64 units.

The loss function was the binary cross-entropy, also known as the log loss:

$$l(y^s, p^s) = -\left(y^s \log(p^s) + (1 - y^s) \log(1 - p^s)\right)$$

where s is a sample in the dataset, $y^s \in [0, 1]$ is the expected output, and p^s is the predicted probability (i.e. the output of the network). We mainly used two optimization methods to minimize this loss: Stochastic Gradient Descent (SGD) [30] and Adam [31]. Because both methods are highly sensible to the learning rate choice, we tried different values, and sometimes applied a rate decay. We also added a Nesterov momentum to the SGD optimizer.

3.4 Results

In order to compare the performance of our classifiers on the three datasets, we needed a robust validation strategy. We chose to validate the models on K-folds cross-validation (CV). The dataset was split into k folds (i.e. subsets), one was used for the model validation, while the $k - 1$ remaining formed the training set. This was done k time, changing the validation fold at each iteration, so that all observations were used once for validation. We chose $k = 10$ folds. Consequently, the 2D dataset (81 patients) had nine folds with eight samples and one with nine. The 3D dataset (85 patients) had five folds with eight samples and five other folds with nine observations.

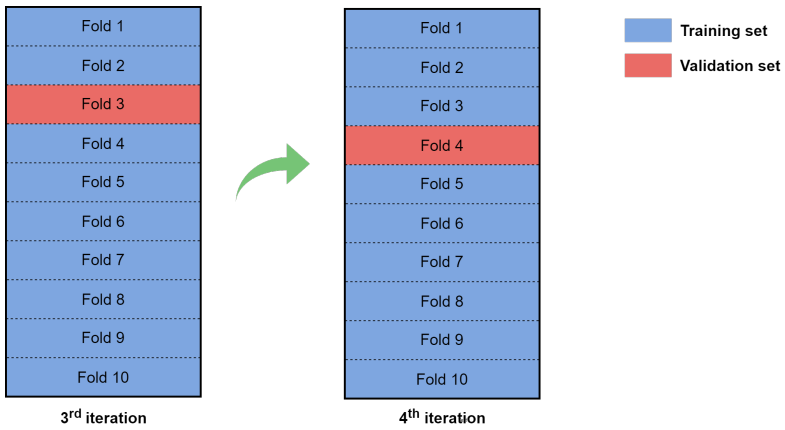


Figure 3.2: K-Fold cross-validation with $k = 10$.

To avoid imbalanced folds, we used `sklearn.model_selection.StratifiedKFold`, that preserves the original representation percentage of each class. In order to use

the same folds for the different classifiers, we set a seed at the definition of the cross-validator. Each iteration provided metrics to evaluate the models performance. Thus, we obtained these metrics 10 times. They were averaged together and summarized in Table 3.1. We computed four metrics: the accuracy, the sensitivity, the sensibility, and the area under the receiver operating characteristics (AUROC).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{TN + FP}$$

where TP, TN, FP and FN are the number of True Positive, True Negative, False Positive, False Negative. The accuracy is the proportion of true predictions among all the samples (without regard to the classes). The sensitivity measures the capacity to well detect all the malignant tumors (positive class). It is also called the recall or the true positive rate. The specificity measures the capacity to well detect all the benign tumors (negative class). It is also called the true negative rate. Finally, we used the AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) to measure the global performance of the model. It tells how much the model is capable of differentiate the classes. A perfect classifier has an AUROC of 1.0, while a random classifier scores 0.5. We also measured the standard deviation of the AUROC (σ_{AUC}) over the 10 folds, to observe the variation of performance between the different training-validation combinations.

Dataset	Classifier	Accuracy	Sensitivity	Specificity	AUROC	σ_{AUC}
2D radiomics	SVM	86.4	83.0	90.0	0.950	0.07
	RF	71.5	75.5	67.5	0.813	0.16
	MLP	79.2	78.5	80.0	0.905	0.08
3D radiomics	SVM	90.8	85.0	97.5	0.984	0.03
	RF	77.8	84.0	70.0	0.818	0.14
	MLP	75.1	76.5	72.5	0.857	0.12
3D radiomics	SVM	100.0	100.0	100.0	**1.000**	0.00
with batch-effect	RF	100.0	100.0	100.0	**1.000**	0.00
normalization	MLP	94.2	93.0	95.0	0.959	0.08

Table 3.1: Validation score for each classifier on the three datasets. Metric scores were computed on 10 CV folds and the mean was taken. σ_{AUC} is the AUROC standard deviation over the 10 folds.

The version of the dataset used by the classifiers has a noticeable impact on the models performance. The 2D radiomics has less features than the 3D dataset (35 vs. 92), that are less informative (e.g. tumor area vs. tumor volume). Moreover, it contains less patient records (81 vs. 85). But somehow, the linear SVM managed to performs well after a feature selection. From the 35 features, the best results were given by keeping only seven of them (*tumor area, tumor perimeter, tumor equivalent diameter, energy, difference entropy, information measure of correlation 1, inverse difference is homon*), and by fitting the linear classifier with a regularization parameter C of 10. The performance on the different validation folds did not vary much. See the ROC curve in the appendix A.4 (page 34). The 3D radiomics improved the SVM and random forest performance, but not the MLP. We make the hypothesis that this was caused by the augmentation of the number of features, adding noise and complexity to the dataset. Although the network should "select" on its own the important features, our dataset has many redundant attributes. In this case, a feature selection might have improved the performance [32]. The batch-effect normalization, that removed the multi-site acquisition variability, had a huge effect on the performance, to the extent that SVM and random forest achieved to classify correctly all the samples of the ten validation folds. The forest had twenty decision trees, and the splits based on Gini impurity. The SVM used a linear kernel, had a regularization parameter C of 1, and was solved using the primal optimization problem with the L1 penalty. We did not find a MLP network that ended with comparable performance. In fact, neural networks allow complex models. Even with regularization and dropout, they tend to overfit quickly when the number of samples is limited. Such datasets need linear and regularized models, like SVM with a linear kernel. The feature selection strategy applied with the SVM classifier performed better that the PCA alternative. The number of principal components were chosen such that 85%, 90%, 95% and 99% of the variance could be explained.

Tumor classification and detection based on MR images

Based directly on the MR images, we tried to classify and detect the tumors. In this part, we compare the different approaches.

4.1 Image preprocessing

Using MR images as inputs for the models is not straightforward. Multiple preprocessing steps are necessary before being able to do so.

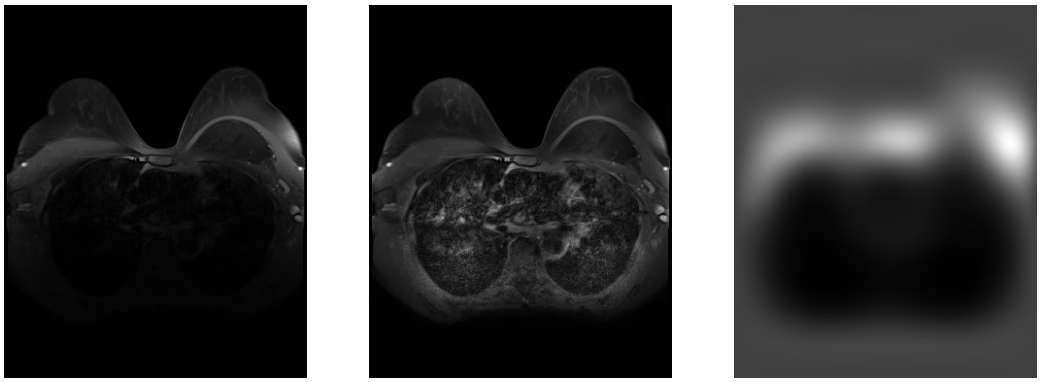
File format

We received the data as a pair of files for each image: the actual image stored in a `.img` file and the associated metadata in a `.hdr` file. This type of storage is inconvenient and not supported by most of the software that we used. A recent file format, and widely supported is called *NIfTI*. It allows storage as a single file, with the extension `.nii` or `.nii.gz` for the compressed version. We used *FMRIB Software Library (FSL)* [33] to convert the files from the original format to the NIfTI one. We created a Python script that iterates through all the files, and make use of `fslchfiletype` command-line.

N4 Bias Field Correction

N4 Bias Field Correction [34] is a popular algorithm, known as *N4ITK*, that corrects inhomogeneity in medical image data, such as low frequency is some parts of the image. Some images of our dataset suffered a lot from this defect, which made the task of

classifying and detecting the tumor harder for the model. We used the implementation provided by *Advanced Normalization Tools (ANTs)* [35]. We found good results setting the number of iterations to 500 and the number of fitting levels to four.



(a) Original image (b) After N4ITK (c) Bias field

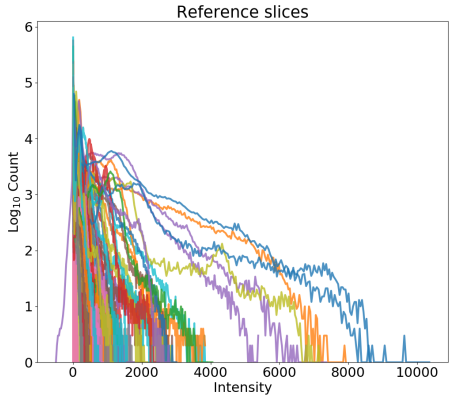
Figure 4.1: Application of the N4ITK algorithm on a 2D slice with a malignant tumor, located in the breast.

Intensity normalization *correcting inter-images intensity variation*

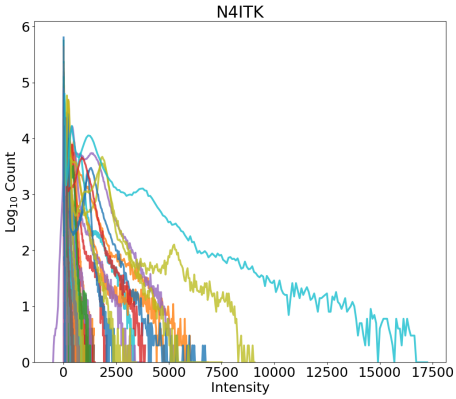
When working on MRI, it is essential to normalize the images in post-acquisition. In fact, MR images do not have a consistent intensity scale. Two acquisitions with the same protocol, the same MR scanner and even the same patient, provide images with potential variations in terms of intensities. Since the models learn to classify and detect the tumors from these intensities, the normalization is imperative. We used two different methods: the Z-score – which simply rescales and shifts the intensities by $I_{new} = (I - \mu)/\sigma$ where μ and σ are the mean and standard deviation of the intensities – and the Nyùl and Udupa method [36] – based on histograms and involving a training and transformation step. These methods are implemented in `intensity-normalization` package [37]. On Figure 4.2, we show the effect of these normalization methods, by plotting the distribution of the pixel intensities, after applying the different techniques.

Bounding boxes

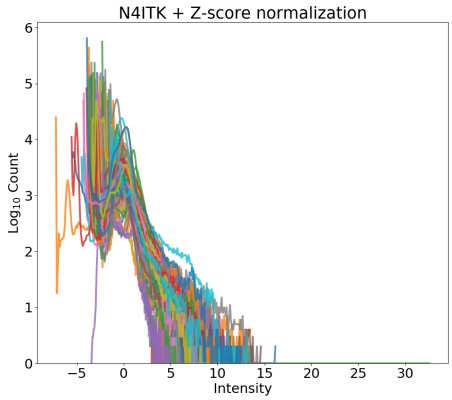
The tumor segmentations were stored in independent files (see Fig 2.3). These files were images (with one or more slices), where only the tumor is visible. The rest of the image is black, encoded by a pixel intensity of 0 over 255. This is called a



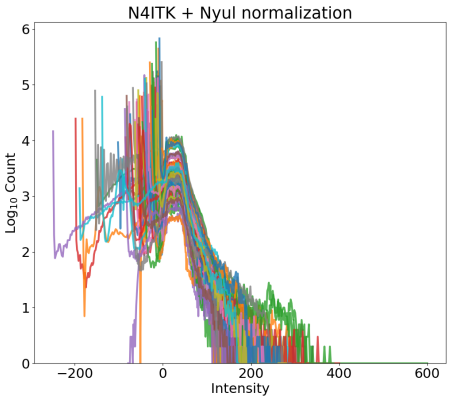
(a) Reference slices



(b) N4 Bias Field Correction



(c) N4ITK with Z-score normalization



(d) N4ITK with Nyùl normalization

Figure 4.2: Distribution of the pixel intensities on the 2D dataset. Each line represents the distribution of a single image.

mask. We developed a short script to retrieve the coordinates of the tumor, indicating its position on the image. These four coordinates allow us to draw a rectangle around the tumor, called a *bounding box*. The coordinates can be the edges of the rectangle $(x_{min}, x_{max}, y_{min}, y_{max})$, or its the center (x, y), width and height. The tumor coordinates of all the images were saved together in a CSV file. By doing so, we compressed considerably the information about the tumor localisation. The masks (i.e. raw images containing the segmentation) took 5GB of memory, while the CSV file needed only 80KB.

4.2 Image classification

Image classification is a computer vision task, where the algorithm receives an image as input, and tries to predict its class. In 2012, this specific task leaded to a revolution in the whole computer vision field, with a tremendous performance gain in the *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*. A new architecture, named AlexNet [38], used deep learning and convolutional neural networks (CNN) to beat the state of the art method. Since then, many new architectures improved this method, surpassing the human-level performance.

In our case, the input of the model could be a full MRI slice, or an image cropped around the tumor. Naturally, the first case is more complex, as the network has to identify the Region of Interest (ROI) (i.e. the tumor) on the full slice before classifying it. Because we had only a small amount of observations, acquired on different body regions, the second case was more adapted. We decided to focus on a classification based only on the ROI.

4.2.1 Custom CNN architecture

With the Keras API, we developed a convolutional neural network (CNN) where we defined each layer. Because the model was learnt from scratch with a small amount of samples, we kept the architecture simple, without too many parameters. The input images were rescaled to a unique size (128×128), and kept as grayscale (one channel). The input shape was therefore $(batch, 128, 128, 1)$. The pixel intensities were normalized between 0 and 1. Three blocks containing a 2D convolution, a batch normalization, a ReLU activation, a max pooling and a dropout were repeated. All the convolutions had a kernel size of 3×3, a padding such that size of the input and output remained the same, and the weights initialized randomly by the Glorot uniform initializer. The first convolution created 16 feature maps (number of channels), the second convolution outputed 32, while the last one produced 64. The pooling size was 2×2, such that the size was divided by two after each max pooling layer. The dropout rate was fixed to 0.5. After the three blocks, the tensor was flatten and followed by a fully connected layer of 32 units, activated by ReLU. A final dropout was placed before the last layer, composed of a single neuron. The latter was activated by the sigmoid function to output a probability. Adam optimizer was used to minimize the

model's loss (binary cross-entropy). The network had approximately 550 000 trainable parameters.

To augment the size of the dataset, we applied some small transformations on the images, so that the network could not see the exact same image twice. They were randomly flipped, zoomed, rotated and shifted. We did not used heavy transformation like shear mapping, which would alter the true shape of the tumor.

4.2.2 Existing backbones

Many state of the art architectures, that have proved to be effective on the ILSVRC, have been publicly released. From these backbones, ResNet [39], Inception-V3 [40] and Xception [41] were considered for our image classification task. We decided to focus on only one of them, ResNet, because all three had approximately the same performance. Unlike our custom CNN, these networks have many layers and millions of parameters. Different sizes of ResNet exist, resulting in shallower or deeper networks. We chose ResNet50, composed of 50 layers and more than 25 millions of parameters. This architecture is already defined in `keras.applications`, and can be loaded directly, providing a `model` with the right layers. However, with such a small dataset and such a deep network, it would be complicated to learn the weights from random initialization. We make use of *transfer learning*. The idea is to load a model that has been pre-trained on a large dataset, and adapt it to our specific task. We used the weights provided by the creator of Keras, François Chollet, on its GitHub[1], that were pre-trained on ImageNet. The last layers, specific to the classification of ImageNet, were removed such that the model output was the final residual block, providing 2048 feature maps of size 7×7. That is a shape of $(batch, 7, 7, 2048)$. On top of the latter, we added new layers: a 2D global average pooling – giving a flat shape of $(batch, 1, 1, 2048)$ – and one or more blocks composed of a fully connected layer, a batch normalization, a ReLU activation and a dropout. As usual, the final layer (i.e. output of the model) was a single unit, activated by the sigmoid function.

The model was *fine-tuned*. We froze the pre-trained part of the network – meaning that the layers parameters were not trainable – such that only our new top layers could update their weights and biases. The network was trained this way during a few epochs. Then, the last block of the pre-trained part was unfrozen, and trained with a small learning rate.

[1] https://github.com/fchollet/deep-learning-models/releases

Most of the pre-trained model receive RGB images as input, and ResNet is no exception. We converted our MR images from grayscale (one channel) to RGB (three channels), by stacking three times each image. This step added complexity and redundant information to our inputs. It is a trade-off between using an existing architecture already pre-trained but with unnecessary complexity, or creating a new model specific to our needs but completely from scratch. A last preprocessing step was to rescale the pixel intensities to the same range that was used to pre-trained the network. This can be done with the `keras.applications.resnet50.preprocess_input` utility function. The dataset was augmented following the same process explained in 4.2.1: applying small transformations on the images.

The gap between our MRI dataset and the ImageNet dataset was large. The pre-trained network has been learned to classify images from the daily life, and was now asked to classify tumors from MRI slices. The transfer should not be a problem if we had thousands of different samples, but it was not the case. We tried to find a larger medical imaging dataset, to use it as an intermediate transfer learning step. After being trained on the latter, the gap with our task would be much smaller. Unfortunately, it is complicated to find large, open and labeled datasets in the biomedical field. In the limited time of the internship, we did not succeed to acquire access to the great BraTS data [42]. However, we found a brain tumor dataset [43] containing 3064 slices from 233 patients, with the associated tumor type labels.

4.2.3 Existing backbones as feature extractor (FE) + ML classifier

A CNN architecture like ResNet could be used to simply extract features. In fact, after the 2D global average pooling layer of a ResNet50, we have 2048 features (see 4.2.2). Normally, in a CNN, we would use fully connected layers on top of the latter, to make our prediction. But we could also use these features as inputs for another classifier, like a SVM. Some papers have shown a performance gain by applying this simple idea [44]. As the task of ImageNet is too far from classifying MR images, we could not extract the features directly with a pre-trained model. We had to retrain this model in order to have meaningful features. We took the weights of our best model so far (i.e. giving the best performance) from section 4.2.2, removed the layers after the 2D global average pooling, and extracted the features for each sample. We chose a tree based ensemble method as classifier, for its robustness to potential irrelevant features.

This allowed us to avoid feature selection. We used the XGBoost (eXtreme Gradient Boosting) [45] classifier, well known for being the state of the art method on many datasets. Three hyperparameters were found using grid search: the learning rate, the maximum depth of a tree, and the number of trees to fit.

4.2.4 Results

The models performance were assessed using the same evaluation strategy than for the radiomics: a cross-validation on ten stratified folds. It is essential to note that the multiple slices from a 3D image were not shuffled on different folds. They all remained in the same fold, so that the network could not be learned and validated on two different slices coming from the same MRI scan. Since the 3D dataset contained more malignant slices than benign ones, we added a *class weight* when fitting the model, to give more importance to each benign observation in the loss function.

Naturally, the training of these models required much more resources than for the radiomics. Indeed, the inputs were vectors for the radiomics, and are matrices or tensors in the image classification task. With all the features included, the largest size for the radiomic vectors was 92, whereas the size of an image for the ResNet architecture is $224 \times 224 \times 3$. Graphics Processing Units (GPU) are more suited than Central Processing Units (CPU) to handle such inputs. The models were trained on an internal server, giving access to *NVIDIA Tesla V100* GPUs. The Python scripts were executed inside a *TensorFlow Docker* container.

Two *callbacks* were used during the training of the CNN (4.2.1) and the ResNet (4.2.2). `ReduceLROnPlateau` was monitoring the last 10 epochs. If the validation loss stopped improving, it divided by two the optimizer learning rate. `EarlyStopping` stopped the training if the validation loss had not decreased during the last 15 epochs, restoring the best weights to the model, and preventing overfitting.

The best results, summarized in Table 4.1, were obtained using a simple Z-score intensity normalization on the original MR images, without a prior N4 Bias Field Correction. The optimal hyperparameters of the N4ITK algorithm were hard to find for each image, and time-consuming. The configurations that we tried improved certain images, but deteriorated the others.

The CNN learned from scratch on the 2D dataset did not succeed to generalize on the validation sets. Too many parameters had to be learned from random initialization,

Dataset	Classifier	Accuracy	Sensitivity	Specificity	AUROC	σ_{AUC}
2D dataset	CNN	∼50.0	∼50.0	∼50.0	∼0.500	∼0.10
(81 patients,	ResNet50	79.9	80.5	79.2	0.878	0.11
81 slices)	FE+XGB	77.7	73.5	82.5	0.833	0.10
3D dataset	CNN	64.6	90.0	10.0	0.531	0.09
(87 patients,	ResNet50	74.1	81.9	57.7	0.800	0.11
2721 slices)	FE+XGB	71.6	80.3	57.0	0.782	0.13

Table 4.1: Validation score for each classifier on the 2D and 3D datasets. Metric scores were computed on 10 CV folds and the mean was taken. σ_{AUC} is the AUROC standard deviation over the 10 folds. *CNN* is the custom CNN presented in 4.2.1, *ResNet50* in 4.2.2 and *FE+XGB* is the CNN feature extraction + XGBoost in 4.2.3

without enough observations. On the 3D dataset, it behaved approximately the same way, giving an AUROC similar to a random classifier. The multiple slices for each patient were not sufficient to train the network.

The ResNet50 model, pre-trained on ImageNet, obtained the best score over all our classifiers based on the MR images. The intermediate training on another medical dataset did not improved this score, probably because the dataset that we used was too small to make a difference. Using this fine-tuned ResNet50 backbone for feature extraction, coupled to a XGBoost classifier to make the predictions, did not improved the performance either.

With the 3D dataset, we used all the slices where the tumor was visible for the validation of the models. For this reason, we observe a decrease in the scores compared to the 2D dataset. Indeed, with the latter only the slice where the tumor was the largest was used in the validation, which is naturally easier to classify than a slice where the tumor is very small, and therefore has less details. It is a trade-off between more data but harder – or even impossible – to classify, or less data but with only the "best" slice for each patient.

4.3 Other approaches tried

4.3.1 Object detection / segmentation

To go further than just classify an image, one could want to detect automatically the tumor on the slice and then classify it. This idea would be possible by solving any of the following tasks: *object detection*, *semantic segmentation*, or *instance segmentation*.

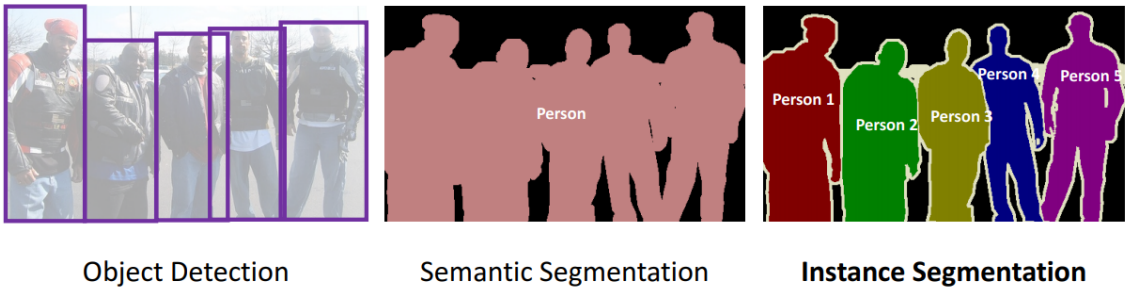


Figure 4.3: Comparison of the different detection tasks.
[https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47]

The object detection task is to draw a *bounding box* (i.e. a rectangle) around each object (e.g. each person on Fig 4.3), and then classify each detection (e.g. say whether it is a person, a dog, or a cat). Unlike object detection, semantic segmentation does not separate multiple instances of the same class, but it classifies each pixel of an image, giving accurate delimitation. Last but not least, instance segmentation delineates the objects as precisely, but by considering them individually, as instances.

Well known methods for object detection are Faster R-CNN [46] and RetinaNet [47] for high precision, YOLO [48] and SSD [49] for quick inference. U-Net [50] and DeepLabv3 [51] are efficient methods for semantic segmentation. Mask R-CNN [52] is the most famous and a powerful method for instance segmentation.

We chose to try the Mask R-CNN approach first, because I already used it personally for a project where it has shown its effectiveness, but also because my code could be reused to prototype quickly to our specific task.

We used the open source implementation of Matterport [53], supporting two backbone network architectures: ResNet50 and ResNet101; we chose the simplest one. The MRI slices were converted to RGB and resized to a unique size. The tumor segmentations were loaded and transformed as binary masks to be used as ground truth. We defined two classes: benign and malignant. The Mask R-CNN model was pre-trained on COCO dataset, a collection gathering "images of complex everyday scenes containing common objects in their natural context" (2.5 million labeled instances in 328k images) [54]. Small data augmentation was applied on the images (flip, rotation, translation, Gaussian blur).

This prototype was developed when we had only access to the 2D dataset. With such a small dataset, learning a complex task like instance segmentation – on samples

that were never seen before – is a challenge. The results were inconclusive, and the detection approaches were set aside, so that we could focus on the radiomics and the image classification. We did not find the time to adapt our code to the 3D dataset and do the training, but we believe that the detections would have been more accurate.

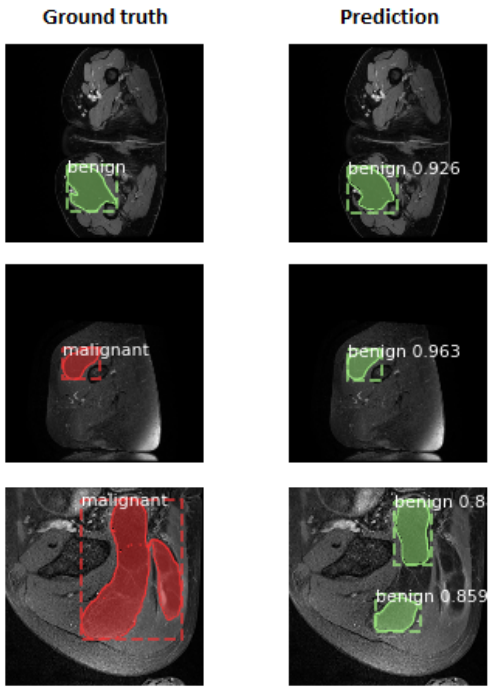


Figure 4.4: Examples of prediction with Mask R-CNN trained on the 2D dataset: a perfect prediction, a good detection but misclassified, and a wrong prediction.

4.3.2 Data augmentation with Generative Adversarial Networks

One of the best way to improve the performance of machine learning models is simply to augment the size of the dataset. With more observations, the models can find better patterns in the data, to fit it more accurately. This is even more typical when working with deep learning methods, which usually outperform traditional machine learning algorithms giving enough data.

During a period of the internship, when we had only one slice per patient (2D dataset), we focused on techniques to augment the size of our dataset. A typical and simple way to do this was to apply transformations on the images – such as flips, rotations, shifts, etc. –, like we did during the training of our CNN, ResNet50 and Mask R-CNN.

But recent papers [55–58] have shown that generating artificial and synthetic images to add new *fake* samples to the dataset, could also improve the models performance in multiple tasks, especially when the dataset is very small or imbalanced.

Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) are two trending classes of generative models. We mainly focused on GANs. The principle behind the latter is fairly simple, two independent networks compete against each other: a generator which tries to generate samples looking real, and a discriminator which tries to determine whether the samples are fake or not. By fighting one another, the networks progressively become robust, generating new realistic samples.

Our first intention was to augment the dataset for the object detection task (4.3.1). To do so, we needed to generate new complete MRI slices, containing a tumor. We also needed the information about the localisation of the synthetic tumor (i.e. its bounding box) with its class to serve as ground truth. A new algorithm called Conditional Progressive Growing of GAN (CPGGAN) was developed to augment a brain MRI dataset for an object detection task [56]. We contacted the first author of this paper, who kindly sent us his TensorFlow implementation of the algorithm. After multiple tries, we did not succeed to get satisfying results on our dataset. Unlike their brain MRI dataset where the slices were very similar, ours contained slices from various body regions. Because our images varied a lot from one another, the generation of realistic slices was much harder.

After this first trial, we focused on a simpler task: generating new tumors, either benign or malignant, for image classification (4.2). A popular and powerful algorithm to generate images with decent quality is called Deep Convolutional Generative Adversarial Network (DCGAN) [59]. Because the dataset contained two tumor classes, we had to train separately a DCGAN on the benign tumors, and another on the malignant tumors. Another architecture called Auxiliary Classifier GAN (ACGAN) [60] incorporates the class condition so that only one GAN could be trained for all the classes. A paper [55] comparing the two methods to augment the observations of a Computed Tomography (CT) dataset, showed better results with DCGAN, so we decided to focus on this approach. Unfortunately, we did not have access to the GPU server yet, and not much time to spend on this data augmentation part, so after some inconclusive prototypes on the *Google Colab* GPUs, it was set aside.

With more time, we could have tried Progressive Growing of GANs (PGGAN) [61] architecture, that can generate high quality and realistic images. We also would have

liked to try Vector Quantised-Variational AutoEncoder (VQ-VAE) [62] that generates samples with large diversity. We would have trained the algorithms on the 3D dataset, by shuffling all the 2D slices.

Conclusion

Based on MRI and machine learning approaches, we achieved to find interesting approaches to differentiate lipoma from ALT/WDL. Using radiomic features, and traditional machine learning classifiers such as Support Vector Machine with feature selection, or Random Forest, we classified correctly all the samples in ten validations folds. This was possible on the 3D batch-effect corrected radiomic dataset. The 3D radiomics improved the models performance from the 2D radiomics. Naturally, the features in the 3D dataset were more informative than with the 2D (e.g. tumor volume vs. tumor area on the largest slice). The batch-effect correction, that removed the inter-site technical variability in the dataset, largely impacted the results. Since the collection came from 43 different centers and 16 particular MR systems, the dataset was highly heterogeneous. Therefore, we can assert that our radiomic-based models generalize well. Nevertheless, they would need to be retested on a new and larger external validation cohort.

In a context of very limited observations, it was a hard task to train models based directly on the MR images. Images are complex inputs, and usually require deep learning architectures, that are fitted by millions of parameters. For these reasons, large datasets are usually required to obtain satisfying results on tasks like image classification or object detection. Using pre-trained models and transfer learning is a possible key to bypass this lock. We believe that the pre-training of models based on medical imaging would be an efficient alternative to well known pre-trained models from daily life image datasets, like ImageNet or COCO. Importantly, unlike many research papers applying CNNs to medical imaging, our MRI collection came from multiple body regions. It is a much harder task to generalize the classification or the detection of tumors located on various organs, due to the high heterogeneity in the images. Moreover, radiomics encapsulated crucial information about the tumor, such as its size. The latter information was not explicitly given to the image-based

algorithms, as the MR images had different zoom levels. The CNN performance might have been higher if the MR slices were set to a unique scale, but we wanted the CNN to find other decision characteristics than the tumor size.

It is important to note that the 3D radiomics depends upon multiple manual segmentations of the tumor for each patient, i.e. on all the slices where the tumor is visible. It is a time-consuming task that is only feasible by experts. Then, radiomic features must be extracted from the MR images and their segmentations. This non-exhaustive pipeline is not an end-to-end solution. A possible end-to-end solution could be using object detection or semantic/instance segmentation methods, that would locate the tumor on the MR image directly, and classify it.

Our exciting results on the radiomics should lead to a scientific publication in a medical journal. We believe that our research could have an impact on clinical practice to differentiate lipoma from ALT/WDL based only on MRI, and in a wider approach, to classify all types of lipomatous soft-tissue tumors.

Appendix

Figure A.1: **Pearson's correlation matrix** of the 3D radiomics with batch-effect normalization.

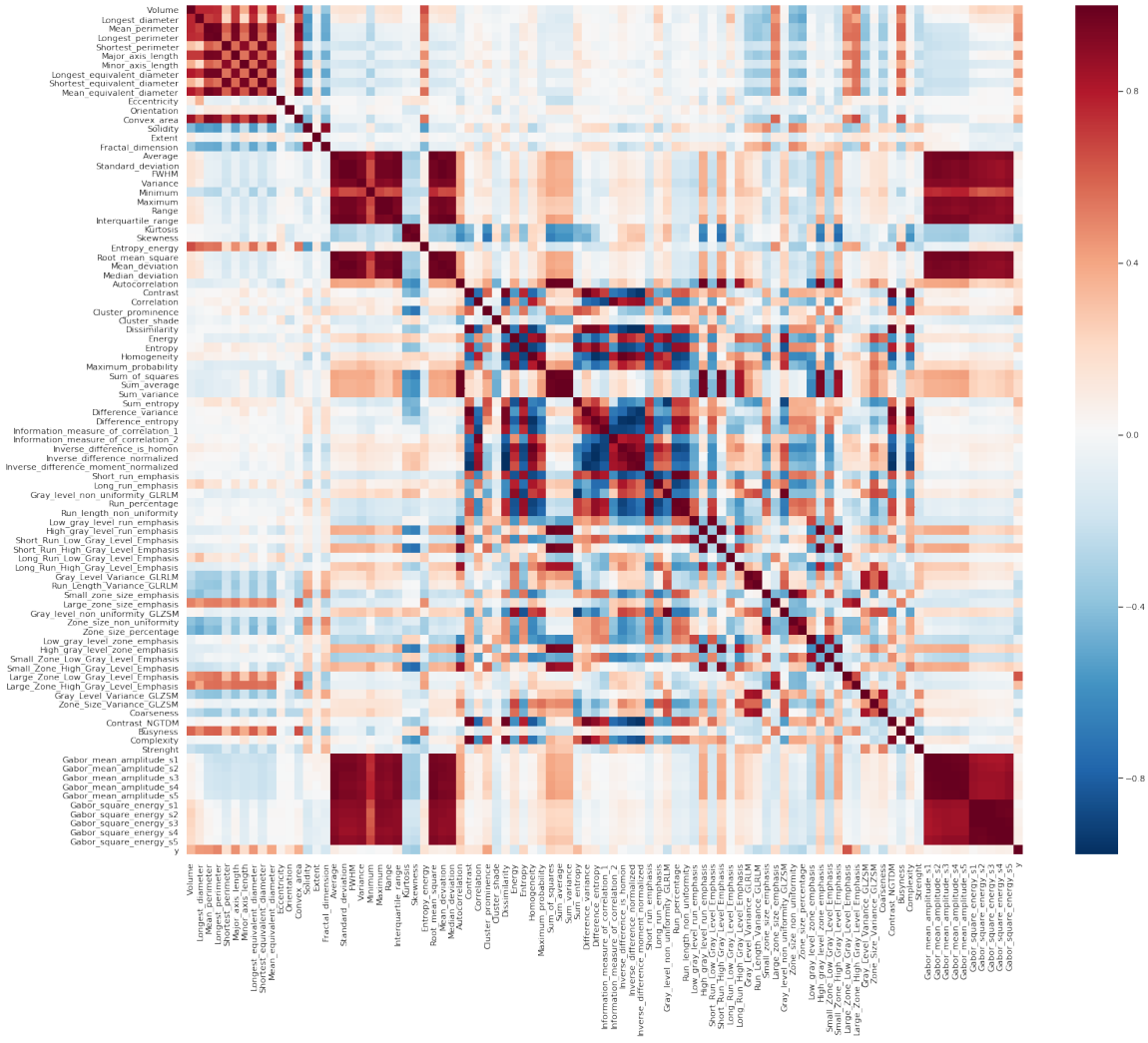


Figure A.2: **Pearson's correlation coefficient between y and the features** of the 3D radiomics with batch-effect normalization.

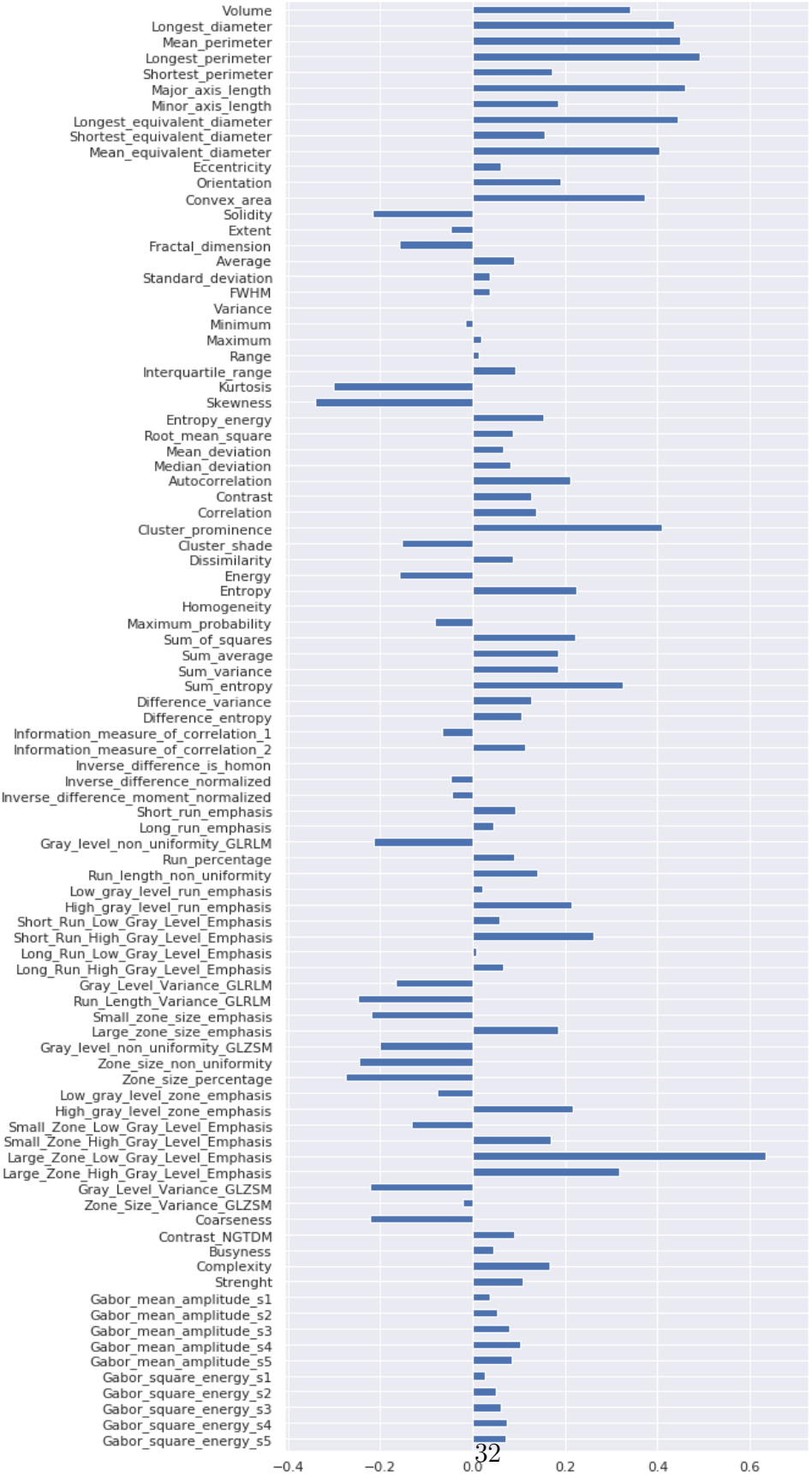


Figure A.3: **MRI slices with tumor segmentation.** For each patient, the chosen 2D slice is the one where the tumor is the largest. In green, the benign tumors (lipomas). In red, the malignant ones (ALT/WDT).

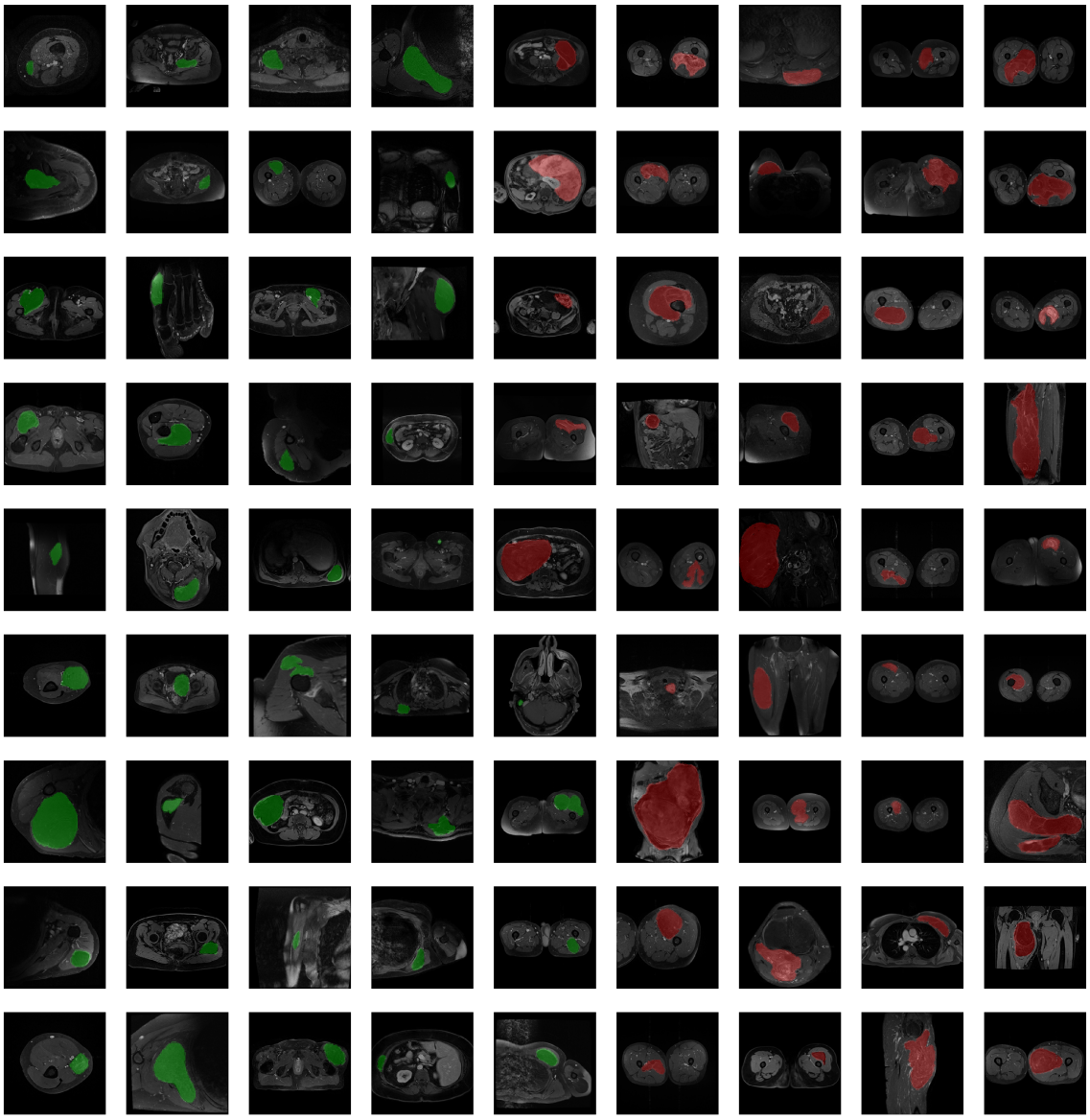
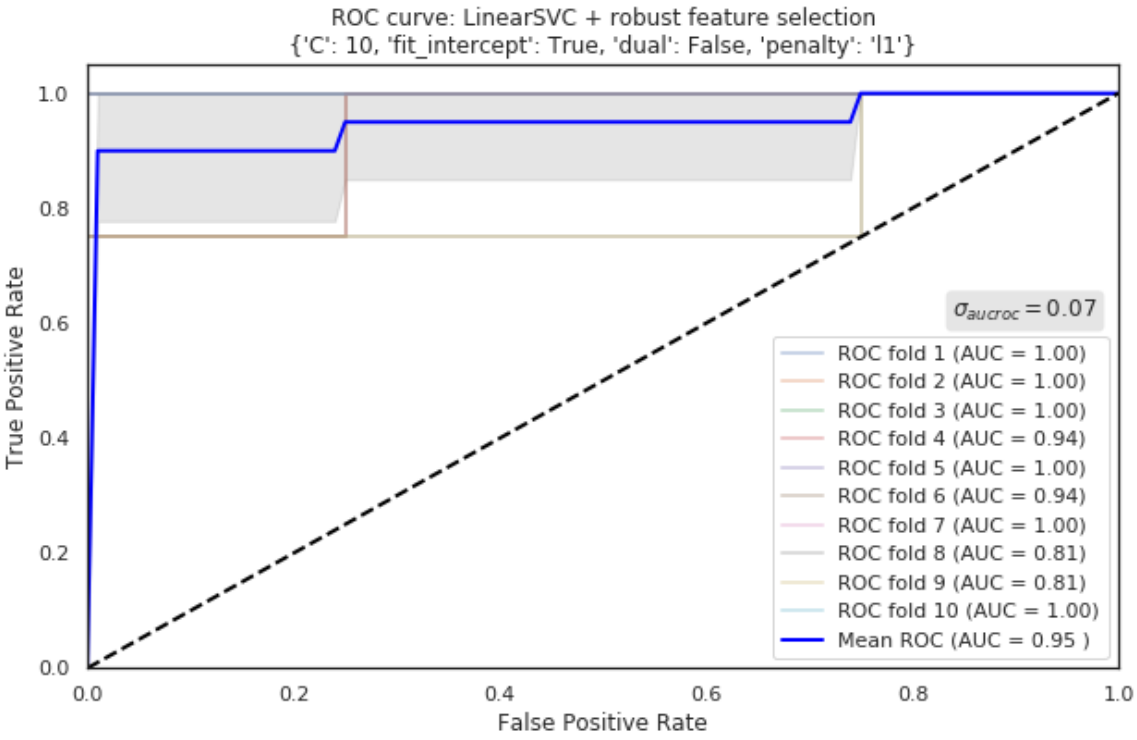


Figure A.4: ROC curve of the best SVM configuration on 2D radiomics



Glossary

ALT/WDL Atypical lipomatous tumor / Well-differentiated liposarcoma. 1–5, 8–11, 13, 29, 30
ANN Artificial neural network. 12
AUC Area Under the Curve. 15
AUROC Area Under the Receiver Operating Characteristics. 2, 15, 24

CNN Convolutional neural network. i, 1, 2, 4, 20–24, 26, 29, 30
CPGGAN Conditional Progressive Growing of Generative Adversarial Network. 27
CPU Central Processing Unit. 23
CREATIS Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé. 2, 4–6
CSV Comma-separated values (file format). 19
CT Computed Tomography. 5, 27
CV Cross-validation. 14, 15, 24

DCGAN Deep Convolutional Generative Adversarial Network. 27

FE Feature extractor. i, 22, 24
FN False Negative. 15
FP False Positive. 15

GAN Generative Adversarial Network. 27
GLCM Gray-Level Co-occurrence Matrix. 6
GLRLM Gray-Level Run Length Matrix. 6
GLSZM Gray-Level Size Zone Matrix. 6, 7
GLSZM Neighborhood Gray Tone Difference Matrix. 6, 7
GPU Graphics Processing Unit. 23, 27

ILSVRC ImageNet Large Scale Visual Recognition Challenge. 20, 21

ML Machine learning. i, 22

MLP Multilayer perceptron. 12, 13, 15, 16
MRI Magnetic Resonance Imaging. 1–5, 9, 10, 18, 20, 22, 23, 25, 27, 29, 30, 33

PCA Principal component analysis. 8, 12, 16
PET Positron Emission Tomography. 5
PGGAN Progressive Growing of Generative Adversarial Networks. 27

RBF Radial basis function. 12
ReLU Rectified Linear Unit. 13, 20, 21
RF Random Forest. 15
RFE Recursive feature elimination. 12
RGB Red, green, blue. 22, 25
RNN Recurrent neural network. 12
ROC Receiver Operating Characteristics. 15, 16, 34
ROI Region of interest. 20

SGD Stochastic Gradient Descent. 14
SVM Support Vector Machine. 2, 11, 12, 15, 16, 22, 34

TN True Negative. 15
TP True Positive. 15

VAE Variational Autoencoder. 27
VQ-VAE Vector Quantised-Variational AutoEncoder. 28

XGB XGBoost (eXtreme Gradient Boosting). 24

References

[1] C. Knebel, U. Lenze, F. Pohlig, F. Lenze, N. Harrasser, C. Suren, J. Breitenbach, H. Rechl, R. von Eisenhart-Rothe, and H. M. L. Muhlhofer, "Prognostic factors and outcome of Liposarcoma patients: a retrospective evaluation over 15 years," *BMC Cancer*, vol. 17, p. 410, Jun 2017.

[2] C. D. Fletcher, K. K. Unni, and F. Mertens, *Pathology and genetics of tumours of soft tissue and bone*, vol. 4. Iarc, 2002.

[3] M. Brisson, T. Kashima, D. Delaney, R. Tirabosco, A. Clarke, S. Cro, A. M. Flanagan, and P. O'Donnell, "MRI characteristics of lipoma and atypical lipomatous tumor/well-differentiated liposarcoma: retrospective comparison with histology and MDM2 gene amplification," *Skeletal Radiol.*, vol. 42, pp. 635–647, May 2013.

[4] H. J. W. L. Aerts, E. R. Velazquez, R. T. H. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, F. Hoebers, M. M. Rietbergen, C. R. Leemans, A. Dekker, J. Quackenbush, R. J. Gillies, and P. Lambin, "Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach," *Nature Communications*, vol. 5, pp. 4006 EP –, Jun 2014. Article.

[5] S. Dodge and L. Karam, "A study and comparison of human and deep learning recognition performance under visual distortions," in *2017 26th international conference on computer communication and networks (ICCCN)*, pp. 1–7, IEEE, 2017.

[6] B. Leporq, A. Bouhamama, F. Lame, C. Bihane, M. Sdika, J. Y. Blay, O. Beuf, and F. Pilleul, "MRI-based radiomic to assess lipomatous soft tissue tumors malignancy: a pilot study," in *International Society of Magnetic Resonance in Medicine and European Society of Magnetic Resonance in Medicine and Biology joint Annual Meeting*, (Paris, France), June 2018.

[7] R. J. Gillies, P. E. Kinahan, and H. Hricak, "Radiomics: Images Are More than Pictures, They Are Data," *Radiology*, vol. 278, pp. 563–577, Feb 2016.

[8] G. Thibault, B. Fertil, C. L. Navarro, S. Pereira, P. Cau, N. Lévy, J. SEQUEIRA, and J.-L. Mari, "Texture indexes and gray level size zone matrix. Application to cell nuclei classification," in *10th International Conference on Pattern Recognition and Information Processing, PRIP 2009*, (Minsk, Belarus), pp. 140–145, 2009.

[9] M. Galloway, "Texture classification using gray level run length," *Computer graphics and image processing*, vol. 4, no. 2, pp. 172–179, 1975.

[10] R. M. Haralick, K. Shanmugam, *et al.*, "Textural features for image classification,"

IEEE Transactions on systems, man, and cybernetics, no. 6, pp. 610–621, 1973.

[11] M. Amadasun and R. King, "Textural features corresponding to textural properties," *IEEE Transactions on systems, man, and Cybernetics*, vol. 19, no. 5, pp. 1264–1274, 1989.

[12] A. Chu, C. M. Sehgal, and J. F. Greenleaf, "Use of gray value distribution of run lengths for texture analysis," *Pattern Recognition Letters*, vol. 11, no. 6, pp. 415–419, 1990.

[13] B. V. Dasarathy and E. B. Holder, "Image characterizations based on joint gray level—run length distributions," *Pattern Recognition Letters*, vol. 12, no. 8, pp. 497–502, 1991.

[14] P. Kickingereder, S. Burth, A. Wick, M. Götz, O. Eidel, H.-P. Schlemmer, K. H. Maier-Hein, W. Wick, M. Bendszus, A. Radbruch, *et al.*, "Radiomic profiling of glioblastoma: identifying an imaging predictor of patient survival with improved performance over established clinical and radiologic risk models," *Radiology*, vol. 280, no. 3, pp. 880–889, 2016.

[15] W. E. Johnson, C. Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical Bayes methods," *Biostatistics*, vol. 8, pp. 118–127, 04 2006.

[16] J.-P. Fortin, N. Cullen, Y. I. Sheline, W. D. Taylor, I. Aselcioglu, P. A. Cook, P. Adams, C. Cooper, M. Fava, P. J. McGrath, M. McInnis, M. L. Phillips, M. H. Trivedi, M. M. Weissman, and R. T. Shinohara, "Harmonization of cortical thickness measurements across scanners and sites," *NeuroImage*, vol. 167, pp. 104 – 120, 2018.

[17] J.-P. Fortin, D. Parker, B. Tunç, T. Watanabe, M. A. Elliott, K. Ruparel, D. R. Roalf, T. D. Satterthwaite, R. C. Gur, R. E. Gur, R. T. Schultz, R. Verma, and R. T. Shinohara, "Harmonization of multi-site diffusion tensor imaging data," *NeuroImage*, vol. 161, pp. 149 – 170, 2017.

[18] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[19] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[20] J.-P. Bergeest and F. Jäger, "A comparison of five methods for signal intensity standardization in mri," in *Bildverarbeitung für die Medizin 2008*, pp. 36–40, Springer, 2008.

[21] N. Cristianini, J. Shawe-Taylor, *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

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

[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[25] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] J. Yang, K. Shen, C. Ong, and X. Li, "Feature selection for mlp neural network: The use of random permutation of probabilistic outputs," *IEEE Transactions on Neural Networks*, vol. 20, pp. 1911–1922, Dec 2009.

[33] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney, *et al.*, "Advances in functional and structural mr image analysis and implementation as fsl," *Neuroimage*, vol. 23, pp. S208–S219, 2004.

[34] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, "N4itk: improved n3 bias correction," *IEEE transactions on medical imaging*, vol. 29, no. 6, p. 1310, 2010.

[35] B. B. Avants, N. Tustison, and G. Song, "Advanced normalization tools (ants)," *Insight j*, vol. 2, pp. 1–35, 2009.

[36] L. G. Nyúl, J. K. Udupa, and X. Zhang, "New variants of a method of mri scale standardization," *IEEE transactions on medical imaging*, vol. 19, no. 2, pp. 143–150, 2000.

[37] J. C. Reinhold, B. E. Dewey, A. Carass, and J. L. Prince, "Evaluating the impact of intensity normalization on MR image synthesis," in *Medical Imaging 2019: Image Processing*, vol. 10949, p. 109493H, International Society for Optics and Photonics, 2019.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition,"

in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015.

[41] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[42] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, pp. 1993–2024, Oct 2015.

[43] J. Cheng, "brain tumor dataset," 4 2017.

[44] Y. Tang, "Deep learning using linear support vector machines," *arXiv preprint arXiv:1306.0239*, 2013.

[45] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, ACM, 2016.

[46] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[48] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[49] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[51] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[52] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the*

IEEE international conference on computer vision, pp. 2961–2969, 2017.

[53] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow." https://github.com/matterport/Mask_RCNN, 2017.

[54] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[55] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[56] C. Han, K. Murao, T. Noguchi, Y. Kawata, F. Uchiyama, L. Rundo, H. Nakayama, and S. Satoh, "Learning more with less: conditional pggan-based data augmentation for brain metastases detection using highly-rough annotation on mr images," *arXiv preprint arXiv:1902.09856*, 2019.

[57] O. Bailo, D. Ham, and Y. Min Shin, "Red blood cell image generation for data augmentation using conditional generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[58] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert, "Gan augmentation: augmenting training data using generative adversarial networks," *arXiv preprint arXiv:1810.10863*, 2018.

[59] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[60] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2642–2651, JMLR. org, 2017.

[61] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[62] A. van den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.