

École polytechnique de Louvain

Heterogeneous Data in Machine Learning and Multimodal Architectures

Use-case applied to tumor segmentation from multiple medical images for radiotherapy

Author: **Nils BOULANGER**

Supervisors: **John A. LEE, Michel VERLEYSEN**

Readers: **Margerie HUET DASTARAC, Ana Maria BARRAGAN MONTERO, Camille DRAGUET**

Academic year 2021–2022

Master [120] in Data Sciences Engineering

Abstract

Artificial intelligence is a field in which much effort is invested to allow machines to understand the world around us. Machine learning models are used to learn from data that describe facts and observations and tend to become better at this thanks to recent advances in this field. The vast majority of them are designed to work with unimodal data, i.e. data coming from one modality, a specific way to acquire information. But the world is far from unimodal and we experience it in a multimodal way through our view, our hearing, our touch, our sense of smell, and our taste. Multimodal machine learning is a branch that aims to exploit multiple sources of information as we do.

The first aim of this thesis is to provide a general point of view on the use of multimodal data in machine learning by introducing five main challenges in this domain, namely: fusion, representation, translation, alignment, and co-learning. It is shown that many fields can take advantage of this approach for all kinds of complex application by showing concrete examples.

The second aim of this thesis is to put the theory into practice by implementing a multimodal approach for a medical segmentation task. This task was segmentation of the gross tumor volume in the head and neck regions for radiation therapy treatment using two types of imaging, medical resonance imaging (MRI) and computed tomography (CT). The novel architecture presented is the X-Net, an adaptation of the well-known U-Net. The X-Net showed promising results compared to its unimodal counterpart and served as a proof-of-concept for its unique design. It also served as a motivation for a more advanced architecture, the nnX-Net, and for more experiences to be performed to analyze the behavior of this network in new scenarios. Finally, several modifications have also been recommended to maximize the potential of the X-Net.

Acknowledgments

Working on this thesis was especially pleasant because I was surrounded by very knowledgeable people who were always ready to help and share their knowledge.

For this, I would like to say a big thank you to Ana, who was always available to help me, especially in the most critical moments of this work. I sincerely think that without you I would not have made it in time. Your very pertinent comments also helped to make this work a bit more qualitative. I must also admit that you made me laugh a lot and I thank you for that.

I also thank Margerie who took a lot of time to help me with the nnX-Net code, the use of clusters and the many problems I had during this work. Thank you Margerie for everything you taught me and for all the help you gave me.

Of course, a big thank you to my supervisors, Pr. Lee and Pr. Verleysen, for their valuable advice throughout this work, which, without them, would never have been possible. Thank you Pr. Lee for the always warm welcome in MIRO and for all suggestions for my manuscripts.

I must also thank my parents, not only for this year, but for the many years spent at the university. It wasn't always easy, but I always had their support and they made sure that I had the best conditions possible to study and enjoy university life.

Contents

1	Introduction	1
1.1	Motivations	1
1.2	Objectives	1
1.3	Methodology	1
1.4	Thesis structure	2
2	Theoretical Foundations	2
2.1	Machine Learning	2
2.2	Deep Learning	3
2.2.1	Artificial and Deep Neural Networks	3
2.2.2	Convolutional Neural Networks	6
2.2.3	Structured Probabilistic Models	7
2.3	Modalities In Machine Learning	9
2.3.1	Quantitative Data	9
2.3.2	Qualitative Data	9
2.3.3	Images	9
2.3.4	Video	10
2.3.5	Text	10
2.3.6	Sound	11
2.3.7	Time Series	11
3	Challenges in Multimodal Machine Learning	13
3.1	Fusion	13
3.1.1	Model-Agnostic Approaches	15
3.1.2	Model-based approaches	17
3.2	Multimodal Representation	18
3.2.1	Joint Representations	20
3.2.2	Coordinated Representation	22
3.3	Multimodal Translation	22
3.3.1	Example-based Models	23
3.3.2	Generative Models	23
3.4	Multimodal Alignment	25
3.4.1	Explicit Alignment	26
3.4.2	Implicit Alignment	26
3.5	Co-learning	26
3.5.1	Parallel Data	27
3.5.2	Non-Parallel Data	27
3.5.3	Hybrid Data	28
4	Fusion in Deep Learning	29
4.1	Introduction	29
4.2	Early fusion	29
4.2.1	Data fusion techniques	30
4.2.2	Early fusion in application	31
4.3	Joint fusion	33
4.3.1	Heterogeneous network design	35
4.3.2	Joint fusion in application	35
4.4	Late fusion	36
4.4.1	Aggregation functions	36

4.4.2	Late fusion in application	37
4.5	Discussion on Fusion Strategies	38
5	X-Net: Multimodal Organs Co-segmentation	40
5.1	Medical Context and Motivations	40
5.2	Objectives	41
5.3	U-Net	41
5.4	X-Net	43
5.5	nnU-Net	44
5.5.1	nnX-Net	44
5.6	Methodology	44
5.6.1	Preprocessing and aata augmentation	46
5.6.2	Models parameters	47
5.6.3	Testing	48
5.7	Results and Discussion	49
5.7.1	General observations	49
5.7.2	Special cases analysis	51
5.7.3	Discussion	54
6	Perspectives	55
7	Conclusion	56
A	Results U-Net and X-Net	58

1 Introduction

The understanding of the world in which we live is multimodal because we use our five senses to perceive what surrounds us. Multimodal means that several modalities are used to describe a fact, an event, or an observation. A modality refers to the way in which the information has been extracted or recorded. For example, sound can be recorded analogously on vinyl or digitally on a computer using bits. As we will see in this thesis, in the medical field, several types of imaging can be used to have images that give different information, even though they target exactly the same area. Multimodal machine learning is a branch that aims to exploit this diversity in information, and we define the multimodal data set when the data that make up it have been acquired from different modalities. In theory, it is clear that having several points of view to represent an event allows us to better understand and describe it, but in practice in machine learning, this can be more complicated, and this leads to five challenges related to this task.

1.1 Motivations

Working with multimodal data can provide a more complete representation of an event. A more complete representation can be translated into three types of interaction that the modalities can have between them. First, we find the complementary modalities, where the addition of a new one allows one to have more information to build a representation. Second, there are the supplementary modalities, meaning that the addition of a new modality improves the robustness of the representation by bringing the same information already present but in a different way. The last type is cooperative modalities and is based on the principle that the addition of a modality can help to better extract the information present in the original. This motivates the fact that working with multimodal data can bring about a series of benefits compared to the conventional multimodal approach.

This motivation has been present for a long time but is amplified by the fact that data are increasingly accessible and are also increasingly multimodal. Advances in technology allow for the development of increasingly complex sensors that allow for the collection of more accurate and complete information about an event. In addition to this, there is the progress made in the field of artificial intelligence, and more particularly in deep learning.

However, multimodal machine learning is still an underexplored subject, and much work remains to be done in this field.

1.2 Objectives

This thesis aims to introduce the reader to the use of heterogeneous data in machine learning and, more particularly, deep learning. At the end of this thesis, the reader will have a global vision of what is being done in multimodal machine learning, will know the different issues, and will understand the diversity of possible applications. The goal will also be to show how a deep learning model can be modified to combine several modalities in order to increase its performance for a real task in the medical field.

1.3 Methodology

To do so, this work was done in two parts. The first one is state-of-the-art, based on several large reviews in the field accompanied by a multitude of concrete examples.

The second part is an application of the theory to a concrete case in the medical field. The multimodal approach will be compared to its unimodal counterpart to motivate the need for more work in this area.

1.4 Thesis structure

This thesis is divided into four chapters that follow this introduction. The first chapter is dedicated to some theoretical notion that is key to understanding the rest of the thesis. We will find an introduction to machine learning as well as deep learning and some of its more advanced concepts like convolutional neural networks. Then the main types of modalities that can be found in multimodal machine learning will be presented.

The second chapter describes the five main challenges in multimodal machine learning with several concrete examples of applications.

The third chapter is entirely dedicated to one of the five challenges: fusion. More precisely, fusion in multimodal deep learning will be discussed and will be divided into three categories: early, joint, and late fusion. For each of these types of fusion, two application examples will be presented.

The last chapter is dedicated to the application of the theory to a medical task. It will be presented what is the gross tumor volume (GTV) segmentation and the main deep learning architectures currently used for this application. A multimodal adaptation will be presented and compared to their unimodal equivalent to demonstrate the usefulness of such an approach. The results will be analyzed, and several tracks of improvements will be suggested.

2 Theoretical Foundations

This master thesis will cover many different topics, most of them specific to deep learning. To fully benefit from the content, it is useful to review these different topics. If some of these topics are familiar to you, feel free to skip the section dedicated to this one.

This theoretical introduction will go back to what machine learning is and then branch off to deep learning. In the deep learning part, we will find convolutional neural networks and structured probabilistic models.

Figure 2.1 allows us to quickly situate artificial intelligence in relation to machine learning and deep learning. In artificial intelligence, the data of perception of an environment is exploited in a raw way with the help of a logical language. In machine learning, data will be processed in a more advanced way and we will be interested in the statistical interactions of these data. Finally, in deep learning, we will talk about high-level abstractions because the information extracted from the data will sometimes be indescribable. This is why we often refer to a black box when we want to describe what is happening in a deep learning model.

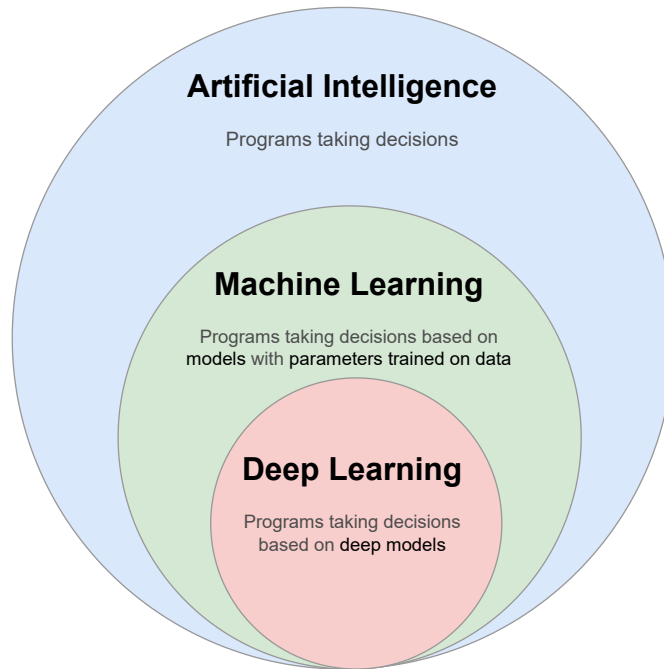


Figure 2.1: Schema inspired by L. Jacques and C. De Vleeschouwer [1]

2.1 Machine Learning

Machine learning is a branch of artificial intelligence in which the goal is to design an algorithm capable of learning from data. In 1977, Mitchell [2] gave this definition of learning: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ." The task is what the machine learning algorithm is asked to do given an example, often decomposed into features containing the information and expressed as a vector $x \in \mathbb{R}^n$ where each entry x_i is a feature. The two main tasks in this area are classification and regression, but there are other applications, such as denoising or anomaly detection.

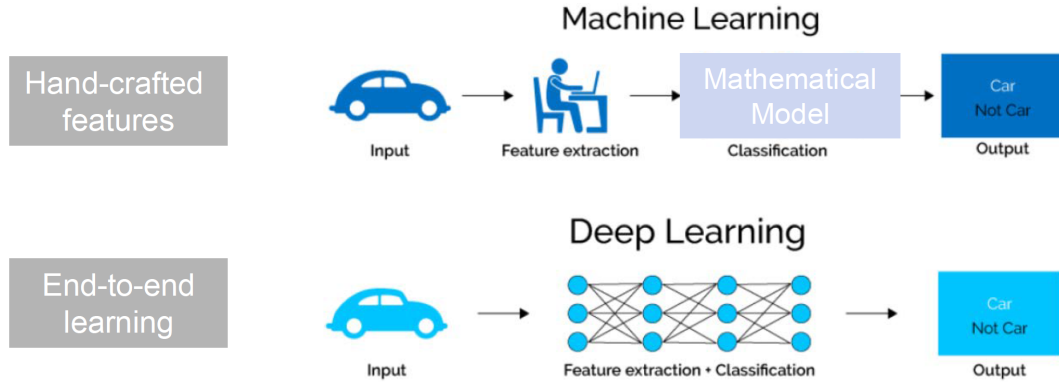


Figure 2.2: Machine learning versus deep learning. Taken from L. Jacques and C. De Vleeschouwer [1]

The performance measure is used to evaluate the learning algorithm on a specific task and depends on the application. Accuracy is probably the most used, since it has many definitions depending on the task. For example, in the case of classification, the accuracy will be linked to the error rate. In fact, there exist many ways to measure the performance of an algorithm, and choosing the best may have a strong impact on learning performance.

In machine learning, the kind of experience available for the learning task has a great deal of influence on the conception of the learning algorithm. The experience is a collection of available examples, commonly referred to as a data set. Depending on the available data, the algorithm is referred to as supervised learning or unsupervised learning. In the first case, labeled data are given and the goal is to be able to predict this label or target for new examples. In the case of unsupervised learning, the focus is on learning useful properties of the structure of the dataset, as in clustering [3]. While most of the time learning algorithms will be used on fixed datasets, it is also possible to design algorithms that can interact with their environment and extract knowledge in this way. This approach is called reinforcement learning and has delivered impressive results in recent years in AI [4].

2.2 Deep Learning

Since the 1990s, the available computing power has never stopped growing and is expected to continue to increase [5], pushing some scientists to develop more and more complex algorithms. In parallel, technological advances allow us to collect more and more data, making the data sets much larger, both in terms of the number of examples and the dimension. Due to the curse of dimensionality, the complexity of a learning task for a given data set is expected to increase as the dimensionality increases [6], and common machine learning approaches have reached their limits with some data sets.

Deep learning is a versatile and powerful field that aims to tackle large and highly complex problems owing to the available computing power. The main idea of deep learning is to draw inspiration from the way the brain works to develop algorithms capable of performing too difficult tasks for traditional machine learning algorithms. This attempt to exploit the pattern of our brain first gave birth to the artificial neural network (ANN), a network composed of several artificial neurons connected by artificial synapses, more commonly called weights. The first introduction of ANNs dates back to 1943 by the neurophysiologist Warren McCulloch and the mathematician Walter Pitts, who tried to model how biological neurons might work together in animal brains using propositional logic [7]. A comparison between machine learning and deep learning can be seen in Figure 2.2.

2.2.1 Artificial and Deep Neural Networks

Before talking about artificial neurons, it is interesting to analyze the structure of the biological neuron from which it is inspired. The neuron is composed of a cell body that contains the nucleus and other

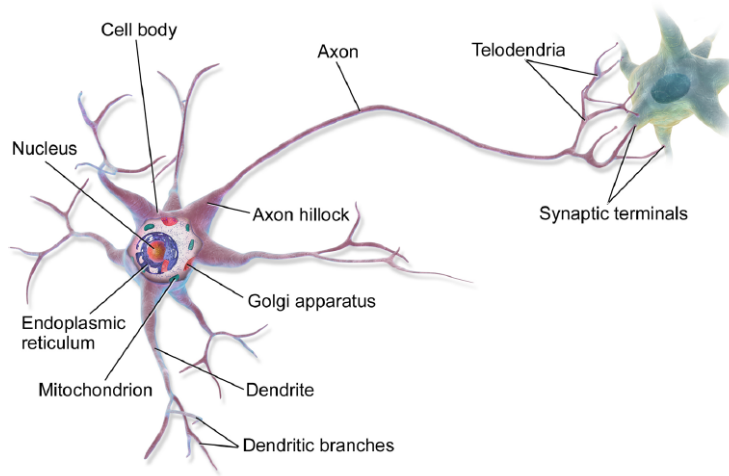


Figure 2.3: Illustration of a biological neuron, taken from Géron [8]

complex components of the cell. Neurons have a branching extension called dendrites and a very long extension called the axon. The end of the axon is composed of telodendria, and these branches are called synaptic terminals (or synapses) which are connected to the dendrites of other neurons. This structure is presented in Figure 2.3. In order to communicate, a neuron produces a short electrical impulse which will travel through the axons. Once a neuron has received a certain number of impulses, it will emit its own electrical impulses. In our body, billions of neurons work together to handle complex computation.

The first artificial neurons, invented by McCulloch and Pitts, is a logical gate that has one or more binary inputs and one binary output (Figure 2.4). Combining different neurons will create a network capable of computing any logical proposition [7].

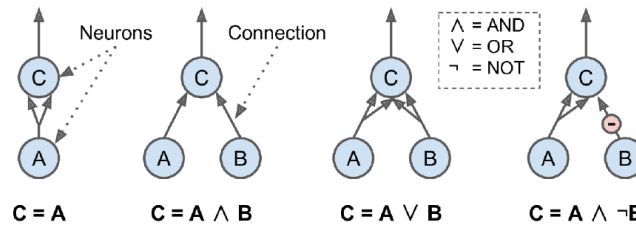


Figure 2.4: Illustration of different artificial neural networks performing simple logical computation, taken from Géron [8]

This model was followed by the perceptron, invented in 1957 by Frank Rosenblatt and composed of multiple logical units. Instead of binary inputs and outputs, these are numbers, and each input will have an assigned weight. In the logical unit itself, a weighted sum of the inputs is calculated, followed by a step function applied to this sum, giving the expected result, as can be seen in Figure 2.5. This is why this unit is called a threshold logic unit (TLU) or a linear threshold unit (LTU).

A single layer of TLUs connected to every input will form the perceptron as shown in Figure 2.6. In the particular case where all neurons in a layer are connected to every neuron in another layer, the layer is called a fully connected layer or a dense layer [8]. A network is composed of three different types of layer. It has one input layer and one output layer, and every layer between these two is called a hidden layer. Even if the definition is not really precise, an artificial neural network having more than 2 hidden layers can be called a deep neural network, and the total number of layers will be its depth. The name comes from the fact that the stack of hidden layers is called deep when there are many of them.

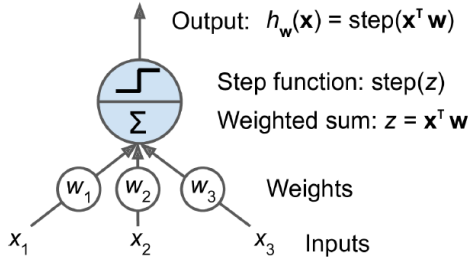


Figure 2.5: Illustration of a threshold logic unit taken from Géron [8]

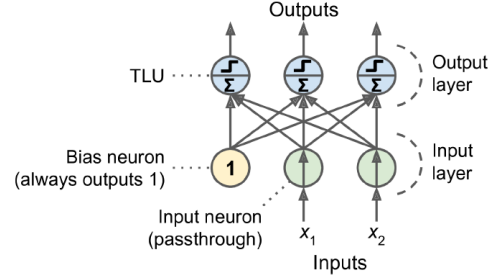


Figure 2.6: Architecture of a perceptron. Illustration taken from Géron [8]

When multiple layers are connected, the network is called multilayer perceptrons (MLPs), also called deep feedforward networks or feedforward neural networks. An example of a multilayer perceptron can be seen in Figure 2.7. We call these networks feedforward because the information flows through the network from the input to the output without feedback provided from the output. A comparison with biological neurons is presented in Figure 2.8.

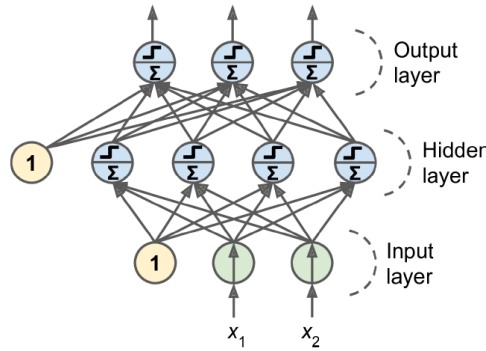


Figure 2.7: Architecture of a Multilayer Perceptron. Illustration taken from Géron [8]

To train such a network, three concepts are used. First, we have the cost function or loss function, which has already been briefly introduced at the beginning of this thesis. The cost function serves as a measure of the performance of the network and is computed by comparing the predictions of the network with the ground truth. The higher the loss function, the less good the prediction is. The most popular cost functions are cross-entropy loss, hinge loss, mean absolute error (MAE) and mean squared error (MSE). At the end of this thesis, another loss will be introduced: the Dice loss. The choice of the loss depends on the nature of the task and, therefore, the type of predictions that need to be done.

The second element we need is the optimizer. The goal of the optimizer will be to minimize the cost function with respect to the weights of the network. Starting from a point on the function, the goal would be to find another point (another set of weights) for which the cost function is smaller. This can be done using, for example, a gradient descent algorithm. In deep learning, the number of parameters can be huge, and a variant of the gradient descent is most commonly used, the stochastic gradient descent (SGD). To train an ANN, the optimizer is used with a technique called backpropagation. In short, the idea is that backpropagation computes the gradient of the loss function with respect to the weights of the network for a single input-output example, instead of doing a direct computation of the gradient with respect to each weight individually. Once the different gradient has been acquired, we can use our optimizer to adjust the weights of the network [8].

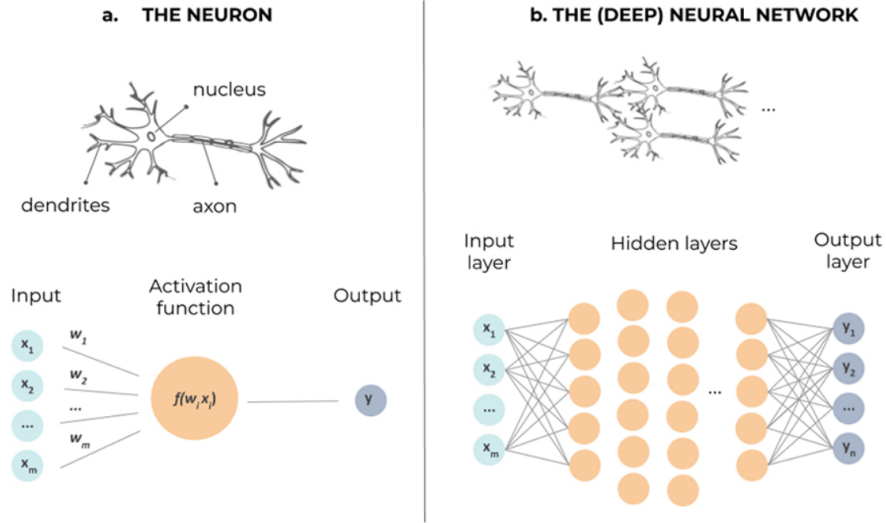


Figure 2.8: Comparison between a biological neuron (a), and a neural network (b). Illustration take from Barragan et al. [9]

2.2.2 Convolutional Neural Networks

Although the deep neural network introduced in the previous section draws its inspiration from the human brain and its neurons, convolutional neural networks (CNN) focus on the visual cortex. Their use dates back to the 1980s, but only recently have they attracted more attention because of the increase in computational power. Although it is used mainly for visual tasks such as handwritten digit recognition [10], organ segmentation [11], or flower classification [12], it is also used for speech recognition tasks [13] or natural language processing (NLP) [14].

The human visual cortex is a part of the brain where there are many neurons with a small receptive field of the scene captured by the eyes, and overlapping these fields will give the whole visual field. There are also neurons that react differently to different patterns in a particular field, such as vertical or horizontal lines. There are also some neurons with a larger receptive field, allowing them to react to more complex patterns [8]. CNNs aim to reproduce these different kinds of neurons through some specific building blocks and a particular operation, called convolution, from which they get their name. A representation of a CNN can be seen in Figure 2.9. A convolutional layer is composed of three elements. A convolution stage, an activation function and a pooling stage.

The convolution stage extract feature of images by doing a convolution of a region of this image with a filter. A filter is composed of different patterns, and if one of the patterns is in the targeted area, the pixel will have a high value for this particular filter. For each pattern, a value will be generated, and we call it a feature. Combining these features creates a feature map which is a representation of the input image in the feature space. Therefore, more features mean better representation of the input image. However, it also means a higher computational cost. Multiple convolutional layers can be stacked to create more abstract representations of the input image. In Figure 2.10, we can observe such a layer where a kernel of 3 pixels by 3 is used. The idea is that applying a filter to a single pixel, i.e. with a kernel of 1 pixel by one, almost no information can be extracted. However, by taking the neighbor pixels, contextual information can be extracted. The larger the kernel, the larger the extracted context. We can also see in Figure 2.10 that a technique called zero padding is used for the border of the image. This is used to ensure that the feature map has the same dimension as the image. When the kernel targets a pixel on the border of the image, pixels with a null value are added to the image. It is also possible to reduce the dimension of the feature map by modifying the stride of the kernel. A stride of 1 means that the kernel will move on the image one pixel at a time. Hence, a stride of two means that the kernel will

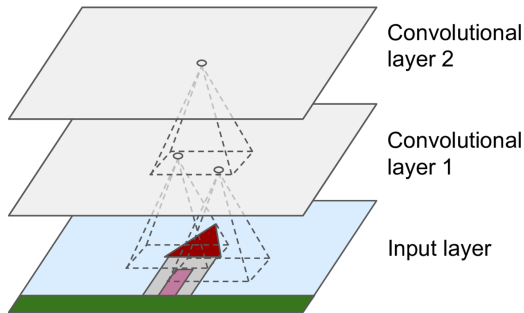


Figure 2.9: CNN layers [8]

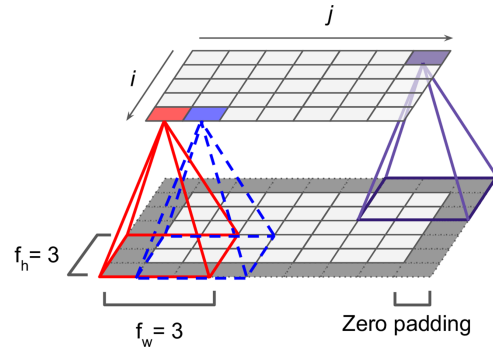


Figure 2.10: Connections between layers [8]

move two pixels at a time, skipping one pixel at each step.

The activation function in the convolutional layer works in the same way as in the one seen before. In CNN, the most used is the rectified linear unit (ReLU).

The last element of the convolutional layer is the pooling operation. It consists of a function that replaces the output of the activation function at a certain location with a summary statistic of the nearby outputs. For example, a max-pooling operation keeps the maximum output within a rectangular neighborhood. The goal of the pooling operation is to make the representation more invariant to small translations of the input [3].

In general, a convolutional neural network will be as follows. First, multiple convolutional layers are stacked, making the image smaller and the number of features larger at each step. Then a regular feedforward neural network is added and is composed of fully connected layers and a final output layer, as seen before. The learning parameters of a convolutional network are the filters of the convolutional layers and the weights of the feedforward network.

2.2.3 Structured Probabilistic Models

Working with data sets involving a large number of variables can be very inefficient (both statistically and computationally) if the goal is to find a probability distribution over all random variables. This is the case if the function used tries to describe the entire joint probability distribution. The idea of structured probabilistic models is that it is possible to represent the global probability distributions with a factorization into distributions over fewer variables since most of the time the probability distribution does not involve direct interactions between all variables.

For example, we try to represent a distribution over three random variables, say a , b and c , and we know that a and c are independent given b . Suppose that a influences b and b influences c . Then we can represent the probability distribution as a factorization into distributions over two variables.

$$p(a, b, c) = p(a)p(b|a)p(c|b) \quad (2.1)$$

This technique, by finding a factorization into distributions over fewer variables, can greatly reduce the cost of representing a distribution because the number of parameters used for each factor is exponential in the number of variables in the factor.

Structured probabilistic models describe this kind of factorization using graphs where the vertices represent random variables and the edges represent direct interactions. Since then, the term graphical models has also been used to refer to structured probabilistic models.

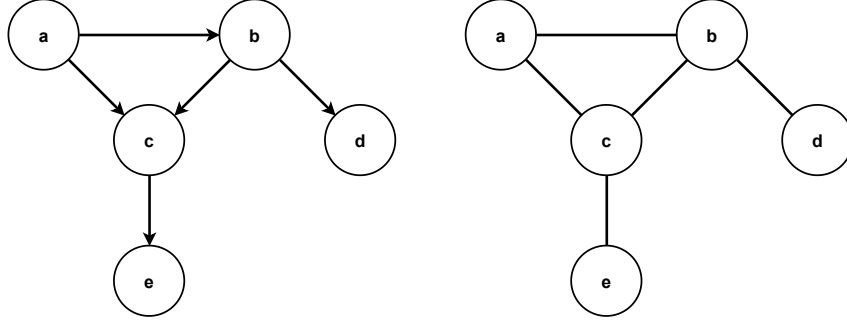


Figure 2.11: Two structured probabilistic models over random variables a, b, c, d and e . (Left) A directed graphical model corresponding to the probability distribution factorized as

$$p(a, b, c, d, e) = p(a)p(b|a)p(c|a, b)p(d|b)p(e|c). \quad (2.4)$$

(Right) An undirected graphical model for the same random variables corresponding to the probability distribution factorized as

$$p(a, b, c, d, e) = \frac{1}{Z} \phi^{(1)}(a, b, c) \phi^{(2)}(b, d) \phi^{(3)}(c, e). \quad (2.5)$$

Redrawn diagram from Goodfellow et al. [3].

The main challenge here is to understand which variables need to be able to interact directly or, in other terms, which graph structure is most suitable for a given problem. To do this, there exist two types of structured probabilistic model [3]. The first is the directed graphical model, also known as the Bayesian network or belief network. The second type is undirected graphical models, also known as Markov random fields or Markov networks. The two types of structured probabilistic model can be seen in Figure 2.11.

In directed graphical models, the conditional probability distribution is represented by directed edges. In the above example, the edge between the vertex a and the vertex b will point to the vertex b . The resulting probability distribution is as follows, where there is a factor for every random variable x_i with parents $Pa_{\mathcal{G}(x_i)}$:

$$p(\mathbf{x}) = \prod_i p(x_i | Pa_{\mathcal{G}(x_i)}) \quad (2.6)$$

Each factor is a conditional distribution of x_i given its parents.

Undirected models do not have directed edges, and the edges represent functions that are usually not probability functions. The factors $\phi^{(i)}(C^{(i)})$ here are functions defined for each clique $C^{(i)}$, where a clique is a set of nodes that are all connected to each other. The normalized factorization of the distribution is written as follows.

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \phi^{(i)}(C^{(i)}) = \frac{1}{Z} \tilde{p}(\mathbf{x}) \quad (2.7)$$

Here, Z is known as the partition function and is defined as

$$Z = \int \tilde{p}(\mathbf{x}) d\mathbf{x}. \quad (2.8)$$

A way to obtain interesting results is to assume that $\forall \mathbf{x}, \tilde{p} > 0$ in undirected models. Using energy-based models (EBM) makes it possible to enforce this condition by taking the exponential of an energy function $E(\mathbf{x})$:

$$\tilde{p}(\mathbf{x}) = \exp(-E(\mathbf{x})) \quad (2.9)$$

This function will always be positive for any $\mathbf{x}$, guaranteeing only a nonzero probability for any energy function, allowing us to learn the energy function with unconstrained optimization. Some of these energy-based models are called Boltzmann machines because the Boltzmann distribution is an example of a function as the previous one. However, there is no specific criterion for when to call a model an energy-based model or a Boltzmann machine.

2.3 Modalities In Machine Learning

In their paper, Lahat et al. [15] defined the term modality as the way information about a phenomenon has been acquired, where various types of sensors can be used under different conditions to extract data. Today, the two most widely used types of data in machine learning are qualitative and quantitative data, but recent advances in deep learning tend to analyze more complex information, such as images, text, or sequential data. In this section, these different modalities are described in detail.

2.3.1 Quantitative Data

Quantitative data store information numerically with continuous or discrete values. Continuous quantitative data do not need a specified number of measurements and can be measured in an infinite scale. It is always possible to have a measurement between two other ones. For example, the amount of fine particles emitted by different vehicles is continuous quantitative data.

Discrete data can only have a finite number of values and are often used for enumeration. For example, the number of children per family represents such a data type.

2.3.2 Qualitative Data

Qualitative data are a non-numerical way to store information mainly about categories, which also gives it the name categorical data. There are several subcategories of qualitative data that are differentiated by the scale used, as defined in [16]:

- Nominal scale
- Ordinal scale
- Interval scale
- Ratio scale

The nominal scale is used for classifications or categories and the variables on this scale do not have a quantitative value. An example of such data is the gender of a person or the breed of a dog.

The ordinal scale provides more than just nominal information; it also provides order in the measurement. Although it is possible to compare the variables with each other, there is no precise distance between them. For example, Bigger/Smaller are ordinal levels or measurements.

The third scale is quite close to the discrete quantitative data, since it refers to measures with equal intervals, also providing information about order. Potdar et al. used in [16] the discrete measurement of temperature as an example, where 1 degree has the same weight anywhere on the scale.

The final type of scale in which qualitative data can be represented is the ratio scale, which combines the qualities of the nominal, ordinal, and interval scales. It has an absolute zero and can be used for comparisons between the values of the variables. In the case of temperature measurement in degree degrees Celsius, the zero value is just another value, but if we use the Kelvin, then we have a ratio scale. Hence, we can say that 600K is twice as hot as 300K, for example.

2.3.3 Images

Images are used to represent a visual context, such as a scene or an object, and provide unique spatial information. The first use of images dates back to 1920, when it was used in telecommunication with the

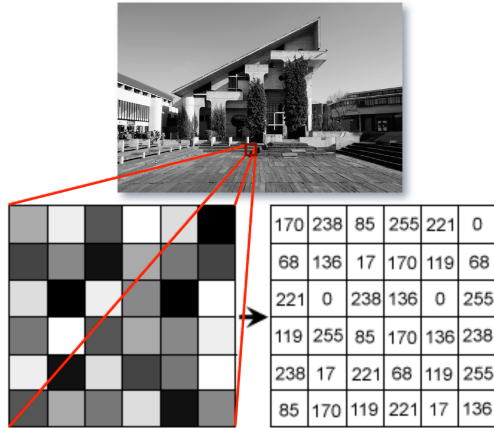


Figure 2.12: A two dimensional image with the corresponding pixel intensity values. Illustration take from [1]

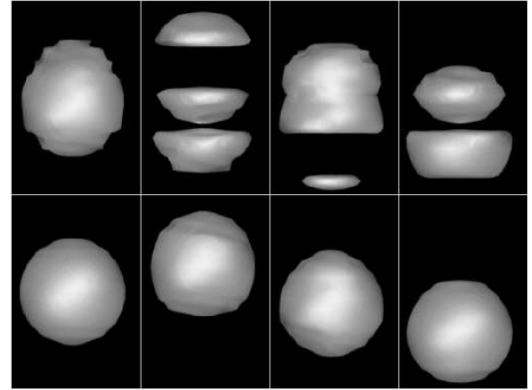


Figure 2.13: Comparison of rendered 3D CT (top) and 4D CT (bottom) of a spherical object periodically moving [19].

Bartlane system [17]. Only in 1969 did images become used in another context, spatial imagery, with the charge-coupled device [18]. It was then followed by medical imaging, digital TV in the 1990s, and today with the Internet, mobile video, 3D, VR [1].

Concretely, an image is a multidimensional signal where each dimension is a physical parameter such as color intensity, electromagnetic radiation, or time, not to mention all of them. In photography, a 2D digital image is a matrix defined by its resolution or the number of pixels in the image and its depth, the number of potential values for each pixel (see Figure 2.12). This is a gray-level image, and to become a color image, one more dimension is needed for the color system, becoming a 3D tensor. As the amount of information with the image increases, the dimension is expected to grow proportionally. For example, this is the case in medical imaging where 4D Computed Tomography is used to study organ motion by incorporating a temporal dimension to obtain a spatio-temporally consistent volumetric image as shown in Figure 2.13 [19].

2.3.4 Video

The video is a sequence of digital images that is used to record a scene in the temporal domain. Each second, a certain number of images are presented, allowing one to give an effect of movement in the scenes. We define this number as frames per second (fps). In cinema, we will typically have 24 fps, which is about the minimum number of fps necessary for the brain not to notice the succession of still images. This modality has all the characteristics of images with a temporal component as well. The difficulty here is not only to extract information from each frame, but also to link them since they are not independent.

2.3.5 Text

A word is a finite set of letters coming from a dictionary of symbols, a sentence is a finite set of words, and a language is a finite set of sentences [20]. Text is used to communicate meaning either through a single word, such as an adjective or an indicator, or through a phrase whose sequence of words has a meaning of its own. In machine learning terms, text is called natural language and its analysis is called natural language processing (NLP). NLP is an automatic process for extracting meaning or concepts from sentences and texts and representing them in a feature/concept space. NLP is generally very complicated and requires a number of capabilities to be successful, as described by Chowdhary et al. [20]:

- Access and acquisition of lexical, semantic, and episodic characteristics
- Creation and propagation of dynamic bindings

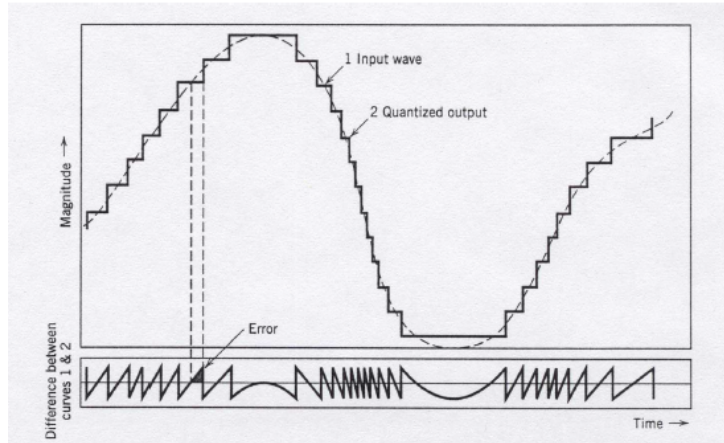


Figure 2.14: Illustration of continuous signal sampling, taken from [21]

- Manipulation of constituent recursive structures
- Coordination of many processing and learning modules
- Identification of basic language constructs
- Representation of abstract concepts

Analysing natural language is complicated because there exist almost infinity of ways to combine words to construct sentences and that the same applies to sentences constructing texts. There is also a lot of ambiguity, and, sometimes, the understanding is unique to each individual. Finally, many words have several meanings.

2.3.6 Sound

Although nowadays sound is used extensively in multimedia and digital content, its origin is much more elementary.

To begin with, sound is the perception of an acoustic wave moving most of the time in the air, but it can also move in any medium, such as a gas, a liquid, or even a solid. The source of an acoustic wave is a vibration that will then propagate. The human ear is designed to be able to perceive vibrations in the air. More precisely, an acoustic wave in the ear will induce a temporal variation in the external pressure in the ear compared to the internal pressure of the ear, allowing the brain to discern a frequency and intensity. The acute sounds will be waves of high frequency, whereas the low sounds will be waves of low frequency. An intact ear is estimated to be capable of capturing a sound wave up to 20 kHz.

In machine learning, the sound will be represented as a sequence of data packets. To get there, the acoustic wave is first recorded with a microphone that will make the transition from a vibration of a membrane to an analog (thus continuous) electrical signal. Then, this signal will be converted into a digital signal using a sampling method followed by a quantization. Sampling consists of transforming continuous values into discrete values, where quantization creates a correspondence between discrete values and bits [21]. The higher the sampling rate, the better the recording quality. An example of sampling can be seen in Figure 2.14. The digital form of sound can be analyzed as a time series presented in the following section.

2.3.7 Time Series

Time series is a term that designates any type of temporal data expressed in the form of a sequence of values or data points, most often spaced in an equal time step. Digitized sound belongs to this category, but we can also find stock market prices, electrocardiograms (ECG), or even a seismic survey.

Most of the time, we will try to make a prediction of the future values of the time series. However, it can also be useful to analyze the contextual value of the information, as, for example, in speech recognition where an audio signal is translated into textual information or expressed in a concept space. The same will be true in the medical domain, where the analysis will focus more on the detection of certain patterns to perform a diagnosis or simply notice an anomaly.

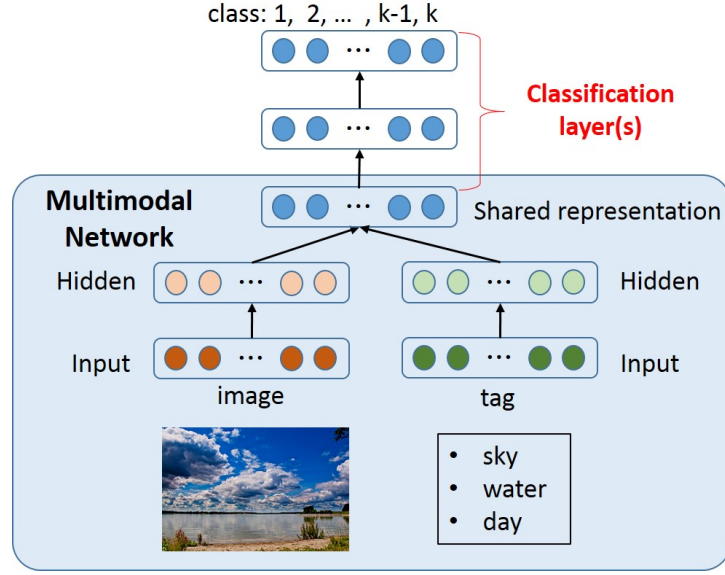


Figure 3.1: Illustration of a multimodal network used to classify images. Illustration taken from Nishida et al. [22]

3 Challenges in Multimodal Machine Learning

Working with heterogeneous data in machine learning has influence at many levels of the learning process. The following chapter will present five different challenges involved in working with multimodal data and will help to have a consistent terminology.

- **Multimodal Fusion:** When talking about working with data from multiple sources, fusion is generally the straightforward approach. Although the different challenges proposed are quite similar, we will define fusion as the combination of data specifically to train a model and make a prediction.
- **Multimodal representation:** This challenge is based on the question of how to represent data coming from different modalities to exploit their complementarity and supplementarity.
- **Multimodal translation:** Translation from one modality to another is another challenge that will be presented. The question raised here is how to translate or map a modality into the representation space of another modality.
- **Multimodal alignment:** The fourth challenge presented is the alignment between heterogeneous data. In a certain sense, we can talk about synchronicity, especially for temporal data. The idea will be to identify which sub-elements of the different modalities correspond to each other.
- **Co-learning** Finally, we will discuss the topic of co-learning. Very close to fusion (it is in a way a sub-branch of fusion), this challenge is different in the sense that the training of a model will be multimodal, but for the prediction, the model will be used in a unimodal way.

3.1 Fusion

When we talk about multimodal machine learning, the first thing that comes to mind is fusion. Not a bad thing, fusion is one of the original research topics when working with heterogeneous data, and is actually one of the most researched aspects of multimodal machine learning [23]. To clearly differentiate fusion from the other challenges presented, we will talk about fusion when the goal is to train a model and

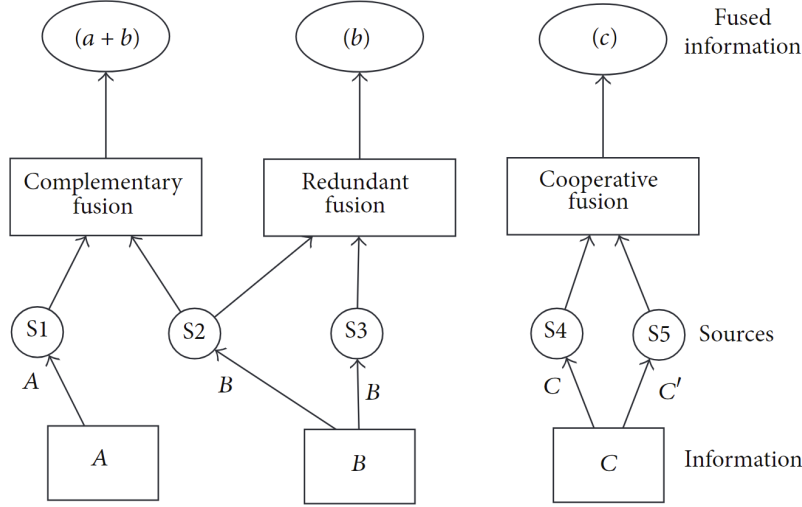


Figure 3.2: Illustration of how data from multiple sensors can enter in relation in a fusion context. Illustration taken from Castenado [25].

predict an outcome using data from multiple sensors. As presented in previous sections, an outcome can be a continuous value through regression or a class through classification. It is worth pointing out that multimodal data fusion is not an area unique to machine learning and affects just about every domain where multiple sensors are used.

The motivations for fusing data from multiple sensors are, in fact, the same as those presented at the beginning of this thesis. Just as having several shots of a scene allows us to better understand what is happening there, having data from different sensors can allow us to describe a phenomenon more completely or in a redundant way. There exist different ways to classify interactions between data from multiple modalities, but the most used is that of Durrant-Whyte [24]. This classification can be seen in Figure 3.2.

In the first case, we will have complementary data. This means that the information provided by one modality is different from that provided by the other, at least in part. Therefore, complementary data can describe a phenomenon in a more complete way. For example, having multiple cameras aiming at a common target from different perspectives will bring complementary information (if the fields of view of these cameras are different, of course). In the case of a model that has to learn a representation of an observation, it will be able to do so with more precision, and thus the predictions that result will be less prone to errors. However, it seems that combining complementary data can increase the risk of overfitting [26], and the machine learning practitioner should be careful about this. Working with such data also means that the fusion techniques used to combine them should be chosen so that the target prediction is not influenced independently by each modality. This point will be covered later, when the fusion techniques are described in detail.

The second way in which data can interact with each other is by being supplementary. This time, the different sensors do not provide data that describe a phenomenon in a more complete way, but in a redundant way. For example, a photography of a scene and the textual description of this scene will provide the same information through two different canals (in reality, this example is not totally correct because the way a scene is described is subjective, and therefore several descriptions for one scene can exist). From the point of view of a machine learning model, this has the advantage of having a more robust prediction when a modality is noisy or even absent. In this case, the other modality can provide the missing information. This type of data will be found when robustness is more important than accuracy in noiseless conditions, as in audio-visual speech recognition (AVSR), where video will be used when the audio signal is noisy, for example.

Finally, the data can be used cooperatively. This means that we will use one or more modality to help us learn another modality. The main idea here is to combine the data that allow new information to be generated, which is generally more complex and does not provide individual information by any modality [25]. This is generally the case where we have multiple modalities when training a model but only one when making the prediction (see section on co-learning). These cooperative data can help models better understand how the target modality will influence the outcome. In simpler terms, the two modalities will bring the same information but in a different way, and the information from of them can help to better understand the information from the other.

In the case of the fusion, two approaches are possible. The first is to use a machine learning model that is used with unimodal data. This means that the multimodal aspect is not part of their basic design. We will speak here of a model-agnostic approach. In the other case, fusion is used with architectures designed to be multimodal. In this case, we speak of a model-based approach. The case of deep learning is a bit special because deep neural networks offer a lot of flexibility in their architectures. This is why we prefer to classify them in the second category because although some basic architectures are unimodal, they can easily be modified to be multimodal.

In machine learning, there are also 3 types of fusions. The first, early fusion, simply combines the multimodal data before the model learns the resulting joint representation. From this, it is also called feature-level fusion and can also be considered in the problem of multimodal learning (see next section). After that, there is late fusion or decision-level fusion, which consists of combining the predictions of different models trained in a unimodal way. In this case, the final model will be based on the marginal representation of the different modalities. The last type of fusion, which came more recently with the arrival of deep learning, is joint (or intermediate) fusion. Here, the idea is to first have independent models that extract the marginal representation of the different modalities and then combine them to learn the joint representation. Although this approach is the most complicated to implement, it would allow us to exploit the relations between the data coming from different modalities and the information specific to each modality. Since these three types of fusion are the most common way in which multimodal data are used in machine learning, chapters will be entirely dedicated to their study later in this thesis.

3.1.1 Model-Agnostic Approaches

As mentioned above, model-agnostic approaches are based on unimodal models. This is why we will mainly find early and late fusion here. Atrey et al. [27] also proposed hybrid fusion, which consists of combining early and late fusion. The completion of one or more fusions involves answering three important questions.

- *How to fuse?* Depending on the modalities used for the learning task, the choice of the fusion technique will influence its efficiency. As has been said, the data coming from the different modalities interact with each other and with the observed phenomenon in different ways, and how to combine the information they provide must be chosen according to this.
- *When to fuse?* When working with data that have a temporal component, it is important to consider the time when the event should take place. It will also be necessary to manage the problem of time synchronicity between the modalities.
- *What to fuse?* Naively fusing all data from different modalities can have a detrimental effect if the information provided by them is contradictory. The question of whether part of the data is missing and whether other data must be used to replace it should also be answered.

In this section, we will focus on providing answers to the first and third questions, while the second question will be studied in the alignment section.

To answer the question of how to fuse multimodal data, we present different fusion techniques below.

Fusion techniques The aim of the fusion techniques is to try to overcome the different problems encountered during the fusion. It is also necessary to best exploit the advantages conferred by the

combination of multimodal data. The main points of attention will be how the modalities interact with each other, what the different levels of confidence are for each of the modalities, and what contextual information they provide. A detailed study of each of these points is beyond the scope of this thesis.

Different techniques for fusion can be classified into three categories depending on the problem space. The first are rule-based methods that include a variety of basic methods for combining multimodal data, such as, for example, a weighted sum. The second is classification-based methods that aim at classifying the predictions of different models, for example. This means that the results from different models themselves will become data points to be learned. The third is estimation-based methods, which are used to estimate the state of a moving object on multimodal data [27]. This is mainly used in video analysis when, knowing the position of the object in the previous frames, the goal is to predict the position of the object in the next frame.

Rule-based fusion regroups a plethora of functions, most of them based on a weighted scheme. The most used and probably also the simplest one is linear weighted fusion. As the name indicates, each modality will be assigned a weight to adjust its contribution either in the vector resulting from the fusion of the features (early fusion) or for the final decision (late fusion). Let X_i be a characteristic vector of the i -th modality on n in total, and let w_i be the associated weight. The resulting vector I of this weighted combination can be obtained by a weighted sum or a weighted product, as follows:

$$I = \sum_{i=1}^n w_i X_i \quad (3.10)$$

$$I = \prod_{i=1}^n X_i^{w_i} \quad (3.11)$$

Although this method seems relatively simple, finding the ideal weights can be a complex task and can depend on the problem. The basic way to do this is to manually set the weights. More advanced techniques will, for example, estimate the reliability of the different modalities or a measure of their mutual information and adjust the weights accordingly. It is also important to normalize the weights, and several techniques are possible to do this, such as using a sigmoid function or using a max-min method [27].

We will also find in this type of method the majority voting scheme, where the final decision will be the one that has received the most votes from the different classifiers (considered here as voters) and the custom-defined rule methods built to meet the needs of a specific application.

Classification-based fusion methods use techniques used in machine learning to classify multimodal observations or predictions. We will find here, among others, the support vector machines (SVM) which are strongly used in the field of multimodal machine learning, Bayesian inference, the maximum entropy model, and different types of network like those presented at the beginning of this thesis.

For example, SVM can be used in late fusion to detect semantic concepts, as shown in Figure 3.3. In this example, the scores of different semantic information from the different classifiers used for each modality are aggregated into a vector and then passed into an SVM. The idea is that points that are close in the feature space produce similar scores and then represent the same semantic concept [28]. On the other hand, Bayesian networks and deep belief networks are particularly suitable for time series data, where deep neural networks will be preferred when working with high-dimensional problems. Recall that probabilistic graphical models can be divided into two types: directed models and undirected models. Bayesian networks are, in fact, directed models, and deep belief networks (or deep Bayesian networks) are a deep implementation of this approach.

The last category of techniques, estimation-based fusion methods, is mainly found in tasks that involve movement, such as tracking objects. The idea is to fuse several modalities to estimate the position of the moving object. The commonly used methods to achieve this are the Kalman filter and particle filter fusion methods. Kalman filtering is an algorithm that estimates a joint probability distribution of past observations for each time frame to predict future values. Particle filters are a set of Monte Carlo algorithms that can be used in the Kalman filtering to help estimate the state of the observations in each frames.

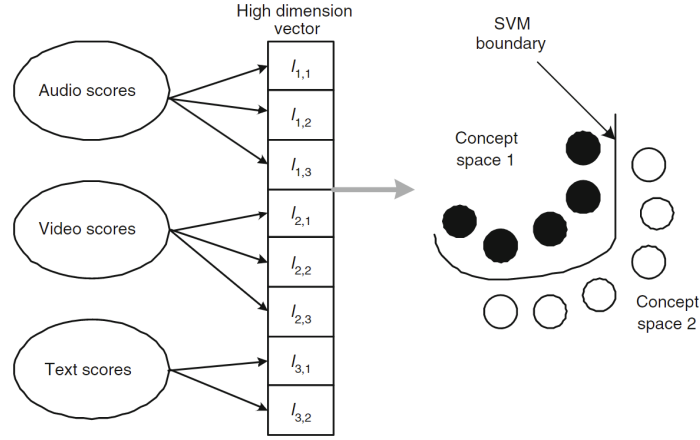


Figure 3.3: Classification of score space based on a SVM to combine information from multiple classifiers. Illustration taken from Atrey et al. [27]

What to fuse To answer this problem, two approaches are possible. The first consists in selecting the modalities considered useful or necessary for the prediction task, and the second consists in performing a feature selection. The second approach is focused on tackling the problem of the curse of dimensionality because concatenating several feature vectors may lead to very large vectors to work with.

Selecting a modality is, in fact, like solving an optimization problem. This could be seen as the sensor stream selection problem, for which the goal is to find the best set of sensors while respecting cost constraints [27]. For example, Wu et al. [29] used independent component analysis (ICA) to identify independent components among the raw feature sets and then grouped these components into several modalities. The goal here was to minimize the dependencies between these modalities and, at the same time, to mitigate the problem of the curse of dimensionality.

The feature selection part and the feature vector reduction are based on known techniques in machine learning, such as principal component analysis (PCA), singular vector decomposition, and linear discriminant analysis (LDA). Several books [30] [31] are devoted to these methods, and the curious reader is invited to refer to them.

3.1.2 Model-based approaches

As said above, model-agnostic approaches are easy to implement because they are based on unimodal machine learning models and methods. Another way to proceed is to develop models specifically for multimodal learning, allowing the best use of the information provided by the different modalities. We will find three main categories of such methods: kernel-based methods, graphical models, and neural networks.

Kernel-based methods This approach has the advantage that kernels can be seen as similarity functions between data points, which allows one to compensate for the fact that different modalities are different representations of a phenomenon. Multiple kernel learning (MKL) methods where the core idea is based on kernel support vector machines. The difference is that different kernels will be used for different modalities. Several kernels can be used, such as linear kernels, polynomial kernels, and Gaussian kernels. There are other kernels for particular applications, such as bioinformatics [32] or natural language processing (NLP) [33]. Gönen et al. [34] emphasize that the selection of the kernel and its parameters is an important matter and recommend the use of a cross-validation procedure to do so. This procedure aims to select the kernel that performs the best among a set of kernel functions by testing them on a validation set. Another advantage of MKL is the fact that model training can be used with standard optimization since the loss function is convex. Finally, this approach is suitable for both classification

and regression. Two main disadvantages with this approach are the fact that deep learning models tend to outperform MKL and the reliance on training data during test time. The consequences of this are a need for a lot of memory and slow inference.

Graphical models As presented at the beginning of this thesis, graphical models provide a graph-based factorization of a distribution with a smaller subset of variables. From their conception, they are particularly well suited for working with multimodal data. Although the section on multimodal representation will speak about the use of deep graphical models, such as deep belief networks, here we will focus on shallow graphical models. The reason why graphical models are particularly useful for the subject of this thesis is their ability to exploit the spatial and temporal structure of the data without making the model too complex [35].

Graphical models can be classified into two categories, generative models and discriminative models. Generative models have received a lot of attention for trying to model a joint probability distribution $p(\mathbf{y}, \mathbf{x})$ on inputs and outputs, especially in statistical NLP. However, this approach has important limitations, such as the fact that the dimension of $\mathbf{x}$ can be very large, but also that the features can have very complex dependencies between them, making factorization very complicated or even an intractable problem. Although the first works that used graphical models to perform multimodal fusion were based on generative models, this approach has lost popularity to discriminative models.

The discriminative approach aims to model the conditional distribution $p(\mathbf{y}|\mathbf{x})$, since it is all that is needed to perform a classification. The main idea will be to combine the advantages of a classification with those of graphical models, i.e., to create a compact output model for $\mathbf{y}$ while being able to handle a large number of input features for prediction [36]. This approach is taken by conditional random fields (CRF), of which a large number of applications can be found in the multimodal machine learning literature [35].

Neural networks Neural networks offer an adequate basis to work with multimodal data and are nowadays widely used for this purpose. The great advantage of working with neural networks is the fact that they can handle a large amount of data, which already gives them a strong point for working with multimodal data. Second, they combine both the multimodal representation task, discussed in the next section, and the fusion task, making the architecture autonomous in a certain way. The last advantage is their performance in general, as specified in the section of this thesis dedicated to them.

However, in general, large training sets will be necessary to obtain good results. In the application part, we will discuss an architecture that allows us to overcome this problem. Finally, a major concern with this approach is the lack of interpretability. It will be difficult to know on which modality the prediction depends and on which modalities play an important role.

3.2 Multimodal Representation

In their book, Bengio et al. [3] defined a good representation as one that makes a subsequent learning task easier. This means that the way information is represented can have a huge impact on the complexity of an information processing task. To illustrate this point, two ways to represent a set of points are presented in Figure 3.4. A common task in machine learning would be to group these data points into clusters. While the first representation (left plot) requires, for example, a density-based clustering technique, the clustering for the second representation (right) is straightforward since it only takes one line to separate the two clusters.

In the above example, it is quite easy to find a good representation for the given task, but in reality, this has always been a challenge in machine learning [35]. In fact, supervised neural network training involves presenting the data to the last layer, often a softmax regression classifier, in a way that it can work with them. This means that the previous layers learn to represent the information in the best way possible. From this, the terms feature and representation can be used interchangeably. A big challenge

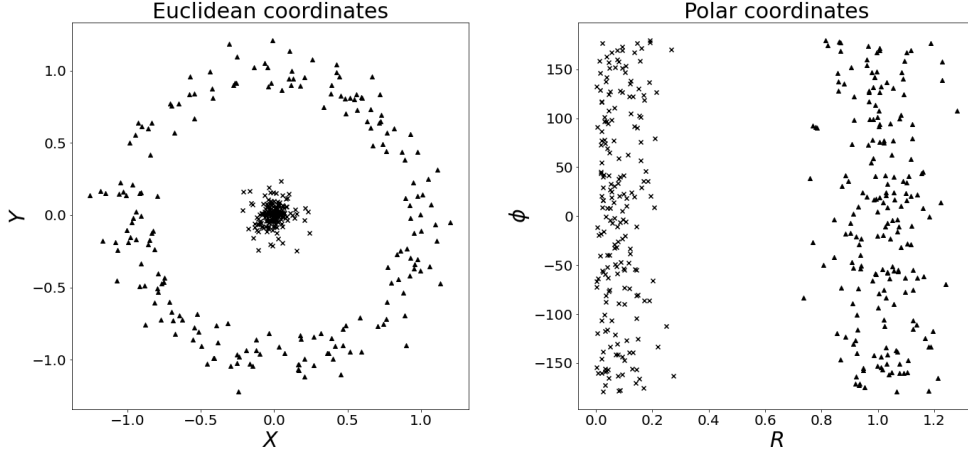


Figure 3.4: The left plot represent some data points in the Cartesian coordinates and the right plot represent these same points in the polar coordinates.

in how to represent data is to find a good trade-off between preserving as much information in the data set and preserving some properties helping the learning task.

In their review on representation learning, Bengio et al. [37] emphasize the fact that representations are a convenient way to express many general priors on data that are not task-specific. Some of these priors are presented here:

- Smoothness¹
- Temporal and spatial coherence
- Sparsity
- Natural clustering

Temporal and spatial coherence means that consecutive observations and nearby observations are related to the same values of the relevant classification concepts or cause small movements on the surface of the high-density manifold. This prior can be applied by punishing changes in values of categorical concepts of interest over time or space, for example.

In mathematics, a sparse matrix is a matrix in which most of the elements are zero [38]. In terms of representation, it means that only a fraction of the factors in the data are relevant and that those that are not relevant are insensitive to small variations in the observed signal or associated features that are often zero [39]. It is easy to see how sparsity can contribute to a clearer representation if the focus is on keeping only the features with key information.

The last prior states that values representing different categories are associated with separate manifolds. In simple terms, values associated with the same category should be grouped.

These criterion have been defined for representation learning, but can also be applied for manual feature engineering. In fact, the real goal of a good representation is to disentangle the underlying explanatory factors hidden in the observed data [37]. This means that the features corresponding to different causes will be represented so that their representation disentangles the causes from each other [3].

When multiple modalities are used, new difficulties arise. The first one is simply how to combine the data coming from different sources, but there is also the question of noise, since they can have different

¹Function to be learned f s.t. $x \approx y$ generally implies $f(x) \approx f(y)$.

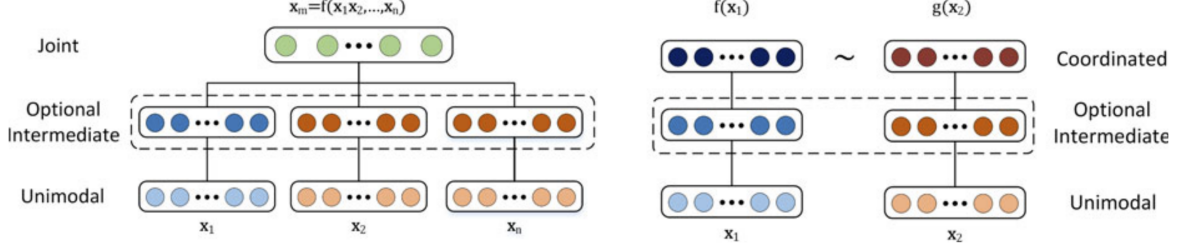


Figure 3.5: Schema of the joint representation (left) and the coordinated representation (right). Illustration take from Baltrusaitis et al. [35]

levels, and the problem of missing data. As a result, the priors mentioned above form a good basis, but more properties are needed to define a good multimodal representation. Srivastava and Salakhutdinov [40] found that the following properties were desirable for a multimodal representation:

- The similarity in the representation space must reflect the similarity of the associated concepts.
- Even in the absence of some modalities, the representation should be easy to obtain.
- Given the observed modalities, it should be possible to fill in the missing one.

Unimodal representation is an area that has been extensively studied to date. For example, a convolutional neural network will create a representation of an image using its convolution blocks. The extracted features then represent the image in the feature space. Recurrent neural networks are also capable of performing the same task for sequential data. However, the study of multimodal representation is a more recent topic of interest to researchers.

Baltrusaitis et al. [35] proposed two categories of multimodal representations. The first is the joint representation, where the different signals are represented in the same space. In fact, this is the kind representation that is constructed when performing an early or joint fusion. The second one is a coordinated representation where each modality is processed separately, but some similarity constraints are enforced between the different modalities, hence the name. These two types of multimodal representation can be seen in Figure 3.5.

3.2.1 Joint Representations

Let x_i be a unimodal representation and $i = 1, \dots, n$. The joint representation x_m of these unimodal representations n can be written as:

$$x_m = f(x_1, \dots, x_n), \quad (3.12)$$

where the function f can be, for example, a restricted Boltzmann machine, a recurrent neural network, a deep neural network, or one of the fusion techniques seen before. Most of the time, we will find this type of representation in tasks where we have multimodal data for both the training and the inference steps. If you remember, it was stated in the previous section that early fusion and multimodal representation are very close, and here we understand why. As we will see in the early fusion section, a simple way to obtain a joint representation is by concatenating the feature vectors of the different modalities. To have a clear distinction with what has been seen and what will be seen on the fusion, we will focus here on more advanced techniques to create representation joints.

We will see three types of architectures that can accomplish this task: the neural networks, the graphical models, and the recurrent neural networks.

Neural networks As we already know, neural networks are capable of performing many tasks, both with a unimodal and multimodal approach. Here, we are interested in the question of how they can be used to create a joint representation. It should be remembered that the basic idea when training a neural

Image	Given Tags	Generated Tags	Input Text	2 nearest neighbours to generated image features	
	pentax, k10d, kangarooisland, southaustralia, sa, australia, australiansealion, 300mm	beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves	nature, hill scenery, green clouds		
	<no text>	night, lights, christmas, nightshot, nacht, nuit, notte, longexposure, noche, nocturna	flower, nature, green, flowers, petal, petals, bud		
	aheram, 0505 sarahc, moo	portrait, bw, blackandwhite, woman, people, faces, girl, blackwhite, person, man	blue, red, art, artwork, painted, paint, artistic surreal, gallery bleu		
	unseulpixel, naturey crap	fall, autumn, trees, leaves, foliage, forest, woods, branches, path	bw, blackandwhite, noiret blanc, biancoenero, blancoynegro		

Figure 3.6: The left part show examples of text generated from a DMB for the input image and the right part show the generated images for some input text. Illustration take from Srivastava et al. [40]

network is to represent the information contained in the data in a feature space. This means that we have taught our network to generate a representation for new data as well, without paying attention to the original prediction task anymore. Moreover, this representation will tend to become more and more abstract if the network becomes deeper. The level of abstraction will grow from layer to layer, that is why we will generally keep the representation in the last layer, which will be the most abstract.

To build the joint representation, we will make an intermediate fusion where the unimodal representations are grouped on a hidden layer, which will project the multimodal information on a joint space. The network can then be trained in a multimodal way and will be able to create a joint representation. This will be seen in detail later, but generally the hidden layer is followed by other layers to build the joint representation and learn from it.

Probabilistic graphical models As we have seen, probabilistic graphical models attempt to obtain a set of latent variables that describe a distribution over a set of observations [37]. Deep Boltzmann machines can be used to represent data where, like in deep neural networks, each successive layer will create a representation at higher level of abstraction. One interesting fact when working with DBMs is that supervised data are not needed for training.

In their work, Srivastava et al. [40] demonstrated that the criterion of a good multimodal representation can be respected with a model based on a deep Boltzmann machine. One of the desired properties mentioned before is in case one modality is missing, it is possible to fill-in it given the observed ones. Figure 3.6 shows an example of generated text and images with a DBM. In this example, a large collection of user-tagged images is used to learn a joint distribution over images and text $P(\mathbf{v}_{img}, \mathbf{v}_{txt} | \Theta)$. Once we have this distribution, which in fact allows us to represent multimodal data using latent variables, we are able to sample either textual information on the image, or images corresponding to the given textual information. In the case of missing textual tags, we can sample them from $P(\mathbf{v}_{txt} | \mathbf{v}_{img}, \Theta)$ and if we want to fill in missing images, we can sample them from $P(\mathbf{v}_{img} | \mathbf{v}_{txt}, \Theta)$.

Deep belief networks, another type of graphical model, are also used in a multimodal context. Kim et al. [41] used such an approach for emotion recognition with audio video inputs. A crucial point for

emotion recognition is to be able to generate meaningful features (or representation) from the data, that is, in multimodal deep learning, to generate a representative joint representation. Deep belief networks were successfully able to generate discriminative features for which the nonlinear dependencies between them were properly approximated.

Graphical models seem particularly suitable to work with heterogeneous data, but there is a catch: the high computational cost. This can become very problematic when working with a large set of data, which is often the case in multimodal machine learning. Hence, it is very important to pay attention to feature selection and dimensionality reduction techniques [35] [41].

Recurrent neural networks Working with sequential data in machine learning often involves recurrent neural networks (RNNs) and, more recently, their variants, such as long short-term memory networks. The idea is the same as the one in neural networks where a hidden layer can be seen as a representation of the data. As we will see, this approach will be particularly useful in joint fusion. We can see the joint representation in a hidden layer of a RNN at a time step t as the summarization of the input sequence up to that time step.

3.2.2 Coordinated Representation

Keeping the notation introduced for the joint representation, the coordinated representation can be expressed as follows:

$$f(\mathbf{x}_1) \sim g(\mathbf{x}_2), \quad (3.13)$$

where f and g are two unimodal mapping functions that create a representation of input data in a feature space. Coordination is expressed through $\sim$, where this can represent a constraint such as minimizing the cosine distance between the two representations, or even maximizing correlation or enforcing a partial order between the two representation spaces [35].

Contrary to the joint representation, the idea here is to create a unimodal representation for each of the modalities and to impose a constraint to make these representations coordinated. Here, two approaches are possible. The first is to create a coordinated representation that supports similarity between the representations. The other one aims to enforce more structure on the resulting space.

The idea of similarity models is that similar concepts from different modalities should be close in the coordinated space. So, for example, the representation of the word dog should be close to the representation of the image of a dog and far from the one of a car [42]. Weston et al. [43] proposed a method called WSABIE aimed at annotating images. The idea is to create a linear mapping between the representation of the images and the words in a way that the inner product between corresponding images and annotations is greater than the non-corresponding ones.

Structured coordinated space models bring new constraints on the representations of the modalities. Most of the time, these constraints will be problem-specific. For example, we can enforce a partial order between information from multiple modalities. This can be done with language et image representation where for example we have an image of a man walking his dog, followed by that text "man walking his dog" and then followed by an image of a man walking.

3.3 Multimodal Translation

Another problem that can arise when working with multimodal data is the translation of information from one modality to another. The simplest example that has already been mentioned in the previous section is the fact of going from a picture to the corresponding text description or vice versa. This problem has been around for a long time, but more recently some interest has surfaced thanks to the work done in natural language processing (NLP) and computer vision. This area is quite broad and often depends on the problem; however, we can distinguish two approaches. The first is based on examples, and the other is generative. These two categories can be seen in Figure 3.7

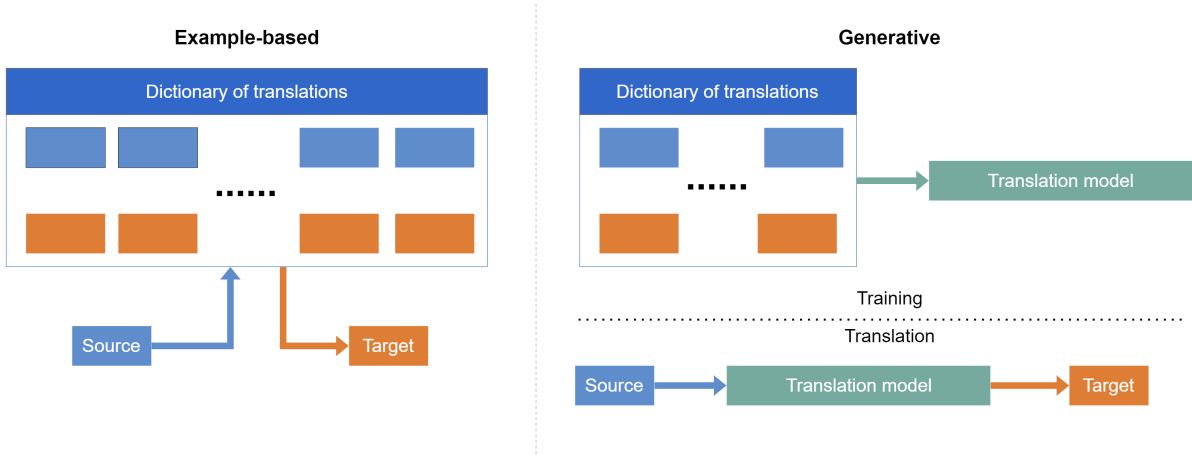


Figure 3.7: Left, a schema illustrating the example-based approach where a dictionary is used to store all know data and compare new data to the existing ones to find the closest values corresponding. Right, schema illustrating of the generative approach where first a model is trained and the available data and then is used for the translation. The main difference is the fact that the output of the generative model should not be a data already seen, unlike in example-based approach. Illustration inspired from Baltrusaitis et al. [35]

3.3.1 Example-based Models

The example-based approach is based on the notion of dictionary, which, in fact, is the training data. In its simplest form, this approach is based on a retrieval-based model where the closest sample in the dictionary, from a semantic point of view, will be the one returned. In order for this to make sense, one must of course work in a feature space or a semantic space so that information from both modalities can be matched. This common space will be called an intermediate semantic space. An advantage of this approach is that it is bidirectional: we can translate one modality to the other and vice versa. However, it will be necessary to manually design this semantic space and have access to many examples to have a complete dictionary.

The other approach is to use a combination-based model, which is an improvement of the retrieval-based model. The idea here is to first combine the examples to provide a better translation. This approach relies on the fact that several descriptions have the same structure, which can then be exploited. For example, Gupta et al. [44] used this approach to automatically generate human-like descriptions of unseen images. The first step is to find neighboring images and rank them according to similarity to the original image. Based on these images and their description, they obtain triples of the form $((attribute1, object1), verb), (verb, prep, (attribute2, object2)), (object1, prep, object2)$, which are then used for the generation of sentences (see Figure 3.8).

Example-based models suffer from a major problem: They rely on the use of an entire dictionary (which, in fact, is the model). Therefore, the model will be very large and the inference quite slow. The other problem is that they are based on the assumption that there will always be a possible translation, that is, that there is always a corresponding value in the dictionary for a given observation. Although the combination approach overcomes some of these problems, it can only be done in one direction.

3.3.2 Generative Models

A generative approach aims to perform a multimodal translation based on a unimodal source that is used as a way to construct an abstract representation that can then be used with other modalities. The difference from the previous approach is that the resulting value is not a real value; it is not in any dictionary. It seems obvious that this approach is complicated. It requires not only understanding the source modality but also generating a target value or sequence. There is also the problem of evaluation

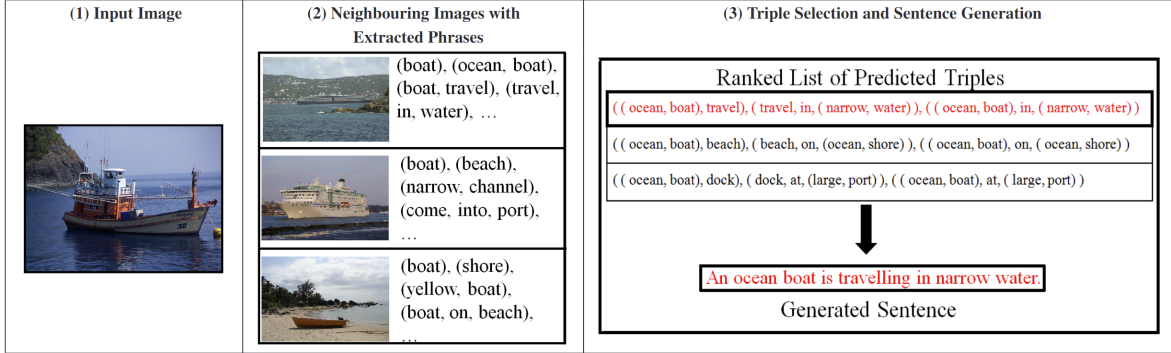


Figure 3.8: Combination-based model for images from the Pascal dataset. Illustration taker from Gupta et al. [44]

because each generated value is unique, so it is not possible to check its accuracy directly. Three categories of generative models can be identified: grammar-based models, encoder-decoder models, and continuous generation models [35].

Grammar-based models Although the idea is to generate a new sequence, it is possible to impose some constraints. For grammar-based models, a grammar can be imposed. For example, to generate a sentence, we can impose that it consists of a subject followed by a verb and then an object.

The first step of this approach consists of identifying high-level concepts in an input observation. This can be objects in an image or actions in a video. These concepts are then combined to generate the target modality. The key here is the combination procedure, where the grammar rules are defined. As a simple example, Barbu et al. [45] proposed a procedure to generate a description of a short video clip in the form of *who* did *what* to *whom*, and *where* and *how* they did it. They used a different classifier to extract the objects and events in the video clip and combined them with the pre-defined grammar.

An interesting advantage of this approach is that it is somehow ensured that the output will make sense because the output must respect a logical syntax. On the other hand, this constraint also limits the diversity of the output. Extracting concepts from a modality can also be challenging.

Encoder-decoder models The idea of an encoder-decoder is to use the decoder to extract the features of a given modality and to use the decoder to express them in the desired modality. This type of model is particularly adapted to the problem of translating from one modality to another. The decoder will often be based on one of the deep learning models seen at the beginning of this thesis in order to extract the features and the choice will be made according to the modality received as input. The encoding part will also be provided by the appropriate architecture for the targeted modality. Most often, RNNs and long-short-term memory (LSTM) architectures will be used [35].

Continuous generation models The last generative approach is continuous generation mainly for sequences. The idea will be to generate an output for each timestep in an online manner. The main application example will be the translation of a video into subtitles like we can have on Youtube. Many approaches are possible for this task like using graphical models, continuous encoder-decoder, or other regression or classification techniques. What really matters here is to remain consistent in time between the two modalities.

Taking the Youtube example, Makino et al. (from Google) [46] proposed a state-of-the-art RNN-based architecture for automatic speech recognition (ASR). The architecture used here is called an RNN-T where T stands for transducer and consists of two components. The first is a transcription network that extracts different concepts from the audio. The second part is a transcription network that will use the

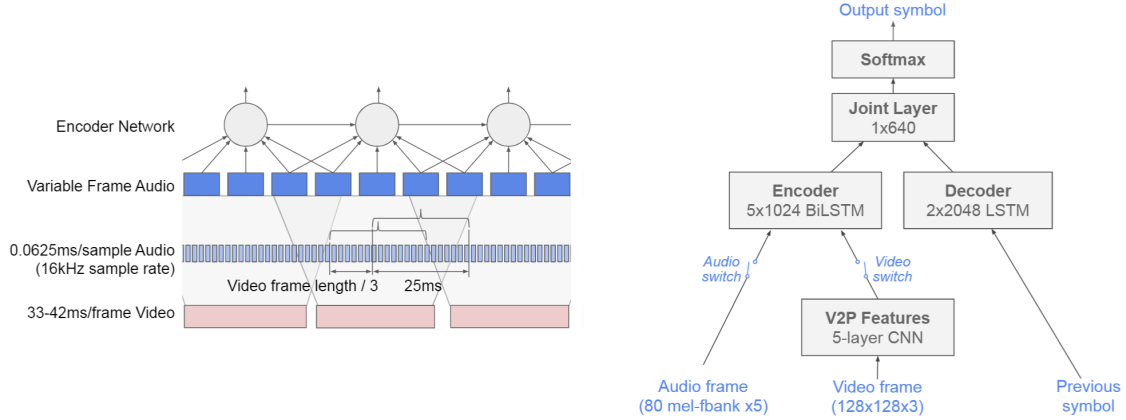


Figure 3.9: Left, an illustration showing how the architecture is used for the ASR problem and how the audio signal is synced with the video signal. Right, the RNN-T model architecture. The video can be added here in order make a video transcription. Illustration taken from Makino et al. [46].

previous generated symbol to model the current symbol. Both components will be joined by feeding them into a joint network. We can compare this approach with an encoder-decoder, and its architecture is presented in Figure 3.9.

3.4 Multimodal Alignment

When working with multiple modalities, it is possible to identify relationships and correspondences between subcomponents of each of them. For example, a textual description of an image will refer to several elements of this image, meaning that each word can be associated with an area of the image. Another rather obvious example is the temporal correspondence between two datasets composed of sequential data where each element must be synchronized at the same timestep.

In the previous example on speech recognition for Youtube videos [46], they had to synchronize the speech signal with the video frames. The real problem is that the sampling rate for both modality is not the same. To overcome this problem, they imposed a certain frame rate for the video signal in terms of frames per second and calculated the corresponding period in milliseconds. They did the same for the audio and created time frames in which the audio signal is incorporated. In order to have a certain continuity in the audio signal, the time frame is a little wider for the audio to have more context information. This can be seen in Figure 3.9.

The ability to align two modalities offers many advantages. One of them, for example, is to be able to assign keywords for each timestep of a temporal signal, allowing us to quickly find a moment in the sequence. In the medical domain, different time measurements can be merged to provide more complete information about the patient. There are many examples where this is interesting; however, several difficulties exist.

First, in the case of supervised learning, there are quite a few datasets where the alignment has been explicitly annotated. Second, it is not easy to design similarity metrics between modalities. Finally, there may either be several possible alignments between the sub-elements of the modalities or there may be no possible correspondence.

We will identify two types of multimodal alignment and the presentation of this challenge will follow this categorization. The first type is explicit, meaning that we explicitly align the sub-elements between the modalities. The other type, implicit alignment, is intended as an intermediary to another task but is not a goal in itself [35].

3.4.1 Explicit Alignment

In this case, it will be particularly important to pay attention to the similarity metric between the subelements of two modalities, which can be defined manually or learned from data. Therefore, two cases can occur. In the first case, we will be in an unsupervised situation, meaning that the alignments are not labeled during the training. On the contrary, in the second case, we will have access to the labels (at least in part).

In supervised alignment, we train a model with multimodal data whose alignments are given, and we must be able to give these correspondences in new situations. Although a lot of work has been based on models working with unlabeled data (it is only recently that we really have datasets with labels for alignment), deep learning models are starting to become the standard for these tasks. As an example of application, Zhu et al. [47] worked on a model where they aligned scenes from a movie with parts of the corresponding book (in case the movie is based on a book).

In unsupervised alignment, the task is more complicated. One of the most common approaches is dynamic time wrapping (DTW), which is based on dynamic programming and is used to align time-series. The idea is to create time frames (as in the YouTube example) and compare them with a predefined similarity measure to match them. Another approach combines DTW with a canonical correlation analysis to not only align the subelements of the modalities but also learn a mapping between the modalities in an unsupervised way [35].

3.4.2 Implicit Alignment

This type of alignment will be used mainly in a latent way in another machine learning task. This approach will be used to improve performance when two temporal modalities are used, for example, in speech recognition or machine translation. This alignment is done in an unsupervised way and during the training phase.

Graphical models have been the go-to approach before neural networks became as popular as they are today. The problem with graphical models is that the mapping between the modalities should be constructed manually, which represents a lot of work.

When it comes to neural networks, they are particularly useful for multimodal translation, and even better when a good latent alignment is done during training. We have seen that multimodal translation can be done using an encoder-decoder approach, but if no implicit alignment is done during training, most of the work is done by the decoder module because it needs to properly summarize the input data before translating them. To address this problem, a mechanism called *attention* [48] can be used, allowing the decoder to focus on the subelements of the input modality. The attention module will recognize the zone of the input observation that should be targeted and translated in another modality.

3.5 Co-learning

The last challenge addressed in the context of multimodal machine learning is co-learning. Co-learning aims to use a resource-rich modality to help modeling a resource-poor modality. A resource-poor modality will be one that has a lack of annotated data, noisy input, or unreliable labels. The use of a resource-rich modality should compensate for these deficiencies during training, but access to this modality is not expected to be provided during inference. Baltrusaitis et al. [35] identified 3 categories of co-learning applications based on their training resources, which can be seen in Figure 3.10. In the first case, we will work with parallel data, meaning that each observation of one modality is directly linked to an observation in the other modality. In contrast, the second case will be to work with non-parallel data, meaning that no direct links between the observations of each modality exist. The final type of data is hybrid data for which the modalities are linked through a shared modality.

Multimodal co-learning is very interesting in the sense that the complementary information across modalities is exploited in order to influence the training of one modality by the other and it is task

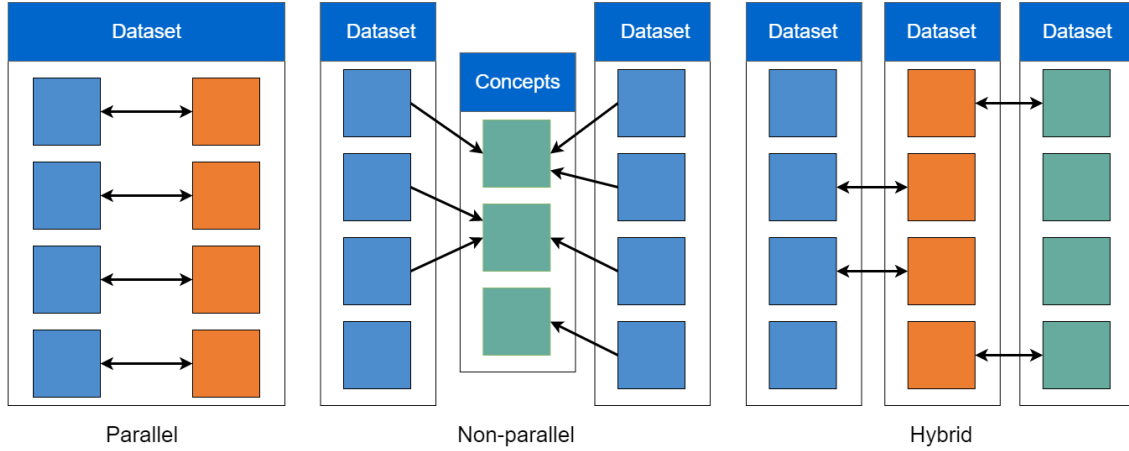


Figure 3.10: Co-learning approaches. Illustration inspired from Baltrusaitis et al. [35]

independent. This approach can also be used to help in other challenges such as for fusion, translation, and alignment.

3.5.1 Parallel Data

As said above, the modalities have their observations that correspond to each other, like the subtitles of a movie that correspond to specific scenes. The idea will be to base ourselves on this to improve the modeling of the modality of interest. Two approaches are possible here: either through co-training [49] (a subcategory of co-learning) where the strong modality is used to generate more labeled data, or through transfer learning [50] where a model transfers the representation of one modality in another.

Co-training is suitable when we have few labeled observations, using the other modality to create more labels. Although this approach is relatively simple, there is a risk of overfitting because biased training samples can be generated.

Based on multimodal representation learning, this approach can use deep Boltzmann machines or autoencoders in the other to transfer the representation learned. In this way, more training samples can be generated in the modality of interest. Only one modality will be needed during inference.

3.5.2 Non-Parallel Data

In this case, instead of having observations that correspond to each other, we will have a correspondence at the level of the concepts extracted from these data for each of the modalities. This is why we will also find the concept of transfer learning here, which is known to be a technique used to train weights of a network on a certain data set and keeping a part of it to train a new model for other data, but in our case we will talk about the construction of a representation instead of training weights. We will also find a conceptual grounding consisting in learning semantic concepts. The last approach is zero-shot learning (ZSL), which consists in recognizing a concept without having encountered it in previous examples.

Even with non-parallel data, we can use transfer learning to learn better representation by transferring the information from the rich modality to the weak modality. This transfer learning approach will be based on coordinated multimodal representations as seen before.

In conceptual grounding, different semantic concepts can be learned from language in the form of text, but also from vision or sound. The idea is to do as the human being works, exploiting as much as possible the perceptual information by drawing from it the concepts describing an observation. For example, Shutova et al. [51] used a grounded representation for visual and linguistic data, which means that the representation is grounded in a common latent space between the modalities. The goal was to obtain a better classification of metaphors and literal language compared to the unimodal approach.

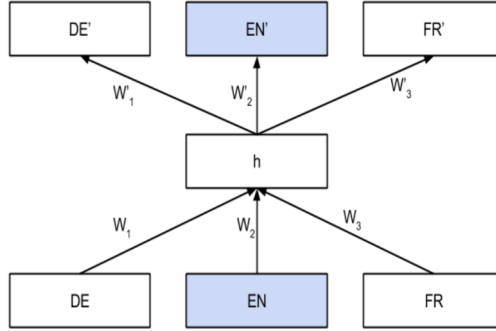


Figure 3.11: Bridge Correlational Neural Network. Illustration taken from Rajendra et al. [52]

They learned the linguistic representations and the visual representations separately, then projected them in a common latent space. In 2016, the term joint fusion did not exist yet, but this approach is much like to belong to this category. More information is given in the section on joint fusion.

The last approach is zero-shot learning (ZSL), consisting in recognizing a new concept, unseen before. For example, recognizing a car without having seen a car example before. This problem is quite challenging but necessary because providing examples for every object that exists is not realistic. In the multimodal case, the targeted object will be identified in the first modality for which the object has been seen, helping identification in the second modality. And then, by definition, multimodal zero-shot learning is only useful for nonparallel data.

3.5.3 Hybrid Data

As said above, hybrid data are based on the concept of modality acting as a bridge between two other modalities. The most famous example of this type of approach has been given by Rajendra et al. [52], where they proposed a bridge correlational neural network (Bridge CorrNets) for multimodal representation learning. Instead of having two views of interest (V_1 and V_2) being parallel data, there is a third view (V_3), called the pivot view, where there are parallel data for each of the other views. To illustrate how this network works, consider the case where the French, German, and English texts are three views of interest, the English view being the pivot (see Figure 3.11). The training data are composed of N_1 English instances with their French equivalent and N_2 instances with this time a German translation. The goal will be to construct and learn a common representation using these three languages, even for French instances for which there is no equivalent in German (in the data set). Bridge CorrNet is based on an encoder-decoder architecture to achieve this.

Another approach consists of using large datasets that are used for different but similar tasks to increase the performance on a dataset that contains only limited annotated data.

4 Fusion in Deep Learning

4.1 Introduction

In this section, the three main types of fusion will be presented in detail, accompanied by a couple of examples. This section is more or less targeted toward deep learning models, but early and late fusion are very general approaches and are commonly used in common machine learning techniques.

The first fusion addressed is the early fusion, for which the multimodal data are fused before being fed into the model. The section will start with some theoretical explanation, then some common early fusion techniques will be presented, and finally some examples will show how this fusion is used.

The second section on joint fusion, an approach in which high-level features are fused, will follow the same structure. First, the theory behind joint fusion will be addressed and different approaches will be presented to implement it. Afterward, some examples will be presented to show how powerful joint fusion is.

The last section will cover late fusion and its applications. Late fusion aims to fuse multimodal prediction for different networks, each working in a unimodal way. Although this last technique of fusion seems relatively simple, the results are really dependent on the choice of the aggregation function.

To conclude this section, a short discussion will be given on the different fusion techniques.

4.2 Early fusion

The first type of fusion discussed is early fusion or feature-level fusion, where the original input data (from multiple sources or sensors) are concatenated into a single vector before being used as input to a machine learning algorithm. Consequently, the term fusion is sometimes replaced with "data fusion" or "multisensor fusion" [23]. From a model point of view, the resulting vector, a joint representation of the input data, will be seen as unimodal and will be threatened as such, meaning that there will be no explicit learning of marginal representations. The concatenation step may also include some preprocessing of the data before they are merged. For example, for some modalities, the features are extracted before merging them with the raw data of other modalities. Huang et al. [53] differentiate between two types of early fusion, where in the conventional case, that is, when only raw data are fused, we speak of type I early fusion, and in the second case, when features have been extracted for certain modalities, we speak of type II early fusion. Both types can be seen in Figure 4.1.

Early fusion could be viewed as an early attempt to perform multimodal representation learning because it allows us to take advantage of the correlation and interaction between the low-level features of each modality [54]. The main reason to choose early fusion as a way to fuse data is simplicity. First, only one model is trained, which makes the process easier compared to intermediate and late fusion, which will be discussed later. Second, no design choices are required on how to extract marginal representation (since it sees the input as unimodal).

However, this simplicity comes with a price. The first drawback is that most of the time, an early fusion of type II is required because it is not possible to fuse the two modalities in their raw form. For example, the sampling rates of various sensors might be different, meaning that the data must first be synchronized before being fused. Another challenge occurs when a sensor produces a continuous data stream, while another produces discrete data. This second problem typically arises when quantitative data need to be fused with qualitative data. Depending on the modality, the feature extraction task can be quite challenging and time-consuming. The second drawback is that, since the model sees the concatenated vector as a unimodal input, it may not be able to identify the relationships between the modalities for higher levels of abstraction [23]. Thus, in a certain way, early fusion models assume that there is no conditional dependence between the states of various sources of information, which is incorrect

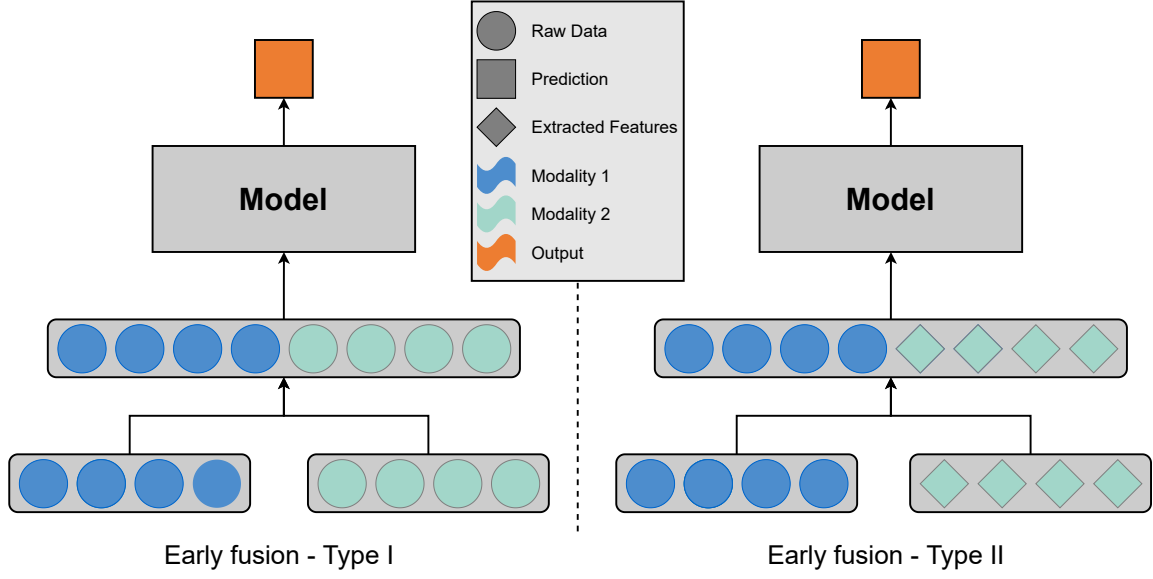


Figure 4.1: Early fusion architecture for type I (left) and type II (right) where the legend indicate the meaning of the shapes and then the meaning of the colors. Schema inspired by Huang et al. [53].

most of the time. Finally, the concatenated vector in early fusion can be very large and, therefore, might contain redundancies. Thus, it will be interesting to apply some dimensionality reduction techniques to eliminate those redundancies.

In their review, Stahlschmidt et al. [26] proposed to make a distinction between early fusion performed for direct modeling, which means that the input vector is fed into a neural network as a unimodal vector, and early fusion done with autoencoders (see Figure 4.2). They have done so because autoencoders behave differently compared to other deep learning architectures, and the learned joint representation can be retrieved for further modeling (see transfer learning). Autoencoders perform well with early fusion because they have a built-in dimensionality reduction capacity, which counteracts the fact that the vector obtained after fusion may have a large dimensionality, especially in the medical domain.

4.2.1 Data fusion techniques

When we speak of early fusion, the word "concatenation" is often used to refer to the fact that the data are fused into a single vector. In reality, it is also possible to fuse input vectors using other strategies [53]. Kiela et al. [55] studied four different continuous approaches to do so:

- Additive model
- Max-pooling
- Gated model
- Bilinear model

Let us have two unimodal feature vectors $\mathbf{x}_u$ and $\mathbf{x}_v$ from two different modalities, where each entry has been normalized to unit length and their associated weight matrices U and V , respectively. Let W be the weight matrix of the network and $o(x)$ the output of the network.

For an additive model, the idea is to perform a component-wise addition of the vectors.

$$o(\mathbf{x}) = W(U\mathbf{x}_u + V\mathbf{x}_v) \quad (4.14)$$

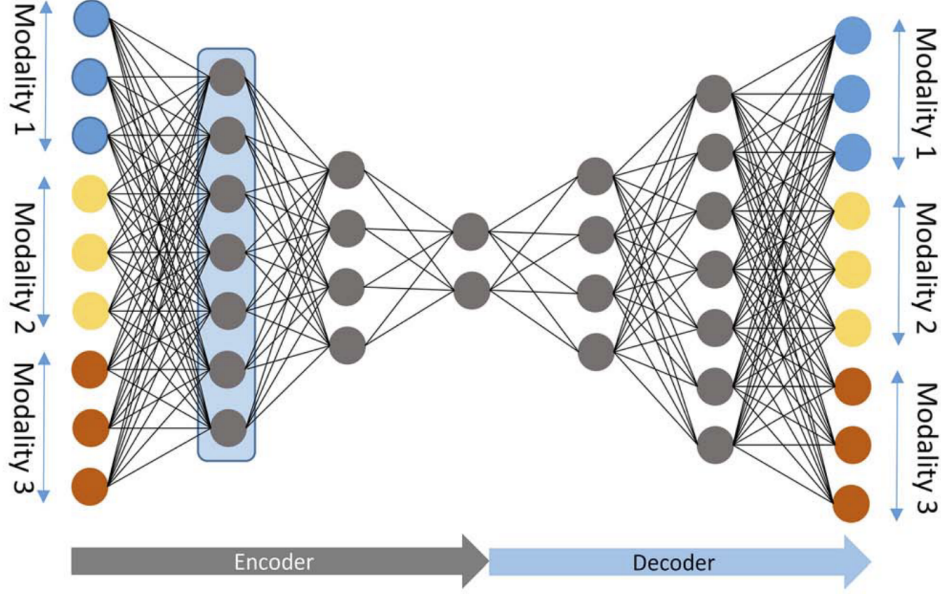


Figure 4.2: Multimodal autoencoder architecture with early fusion. Schema taken from Stahlschmidt et al. [26].

With a max-pooling strategy, the features of both modalities are combined using the component-wise maximum.

$$o(\mathbf{x}) = W \max(U\mathbf{x}_u, V\mathbf{x}_v) \quad (4.15)$$

The gated model is based on the gated multimodal unit presented by Arevalo et al. [56]. For the sake of simplicity, a simplified model is described here. The idea is that for one of the two modalities, a gate σ will control the contribution of the features to the resulting vector of fusion. Figure 4.3 illustrates a bimodal unit where $\mathbf{h}$ is the resulting vector and the activation function $\tanh$ is intended to encode an internal representation of the input feature vectors, but is neglected here. The simplified model is as follows.

$$o(\mathbf{x}) = W(\sigma(U\mathbf{x}_u) * V\mathbf{x}_v) \quad (4.16)$$

The gate σ can be applied to $\mathbf{x}_u$ or $\mathbf{x}_v$.

The final approach uses a bilinear model to fully capture any associations between the two modalities. The $\otimes$ denotes the outer product:

$$o(\mathbf{x}) = W(U\mathbf{x}_u \otimes V\mathbf{x}_v) \quad (4.17)$$

An example of a more sophisticated bilinear model to perform a fusion can be seen in Figure 4.4. The first step is to extract the features of the image and the description of the scene (text data). The image and text representations are then projected into a higher-dimensional space. Lastly, both vectors are convoluted using an element-wise product in the Fast Fourier Transform (FFT) space [57].

It is important to note that this list of techniques is not exhaustive, and there are a number of different ways to combine data. In fact, this problem does not only concern fusion, but is also related to the other challenges in multimodal learning discussed earlier.

4.2.2 Early fusion in application

Multimodal skin lesion classification using deep learning [58] The first application aims to use a multimodal approach for skin lesion classification by fusing patient metadata with multiple imaging modalities. For this purpose, they had at their disposal metadata with macroscopic and a dermatoscopic

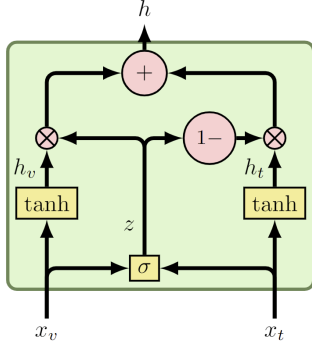


Figure 4.3: A bimodal gated unit. Illustration taken from Arevalo et al. [56]

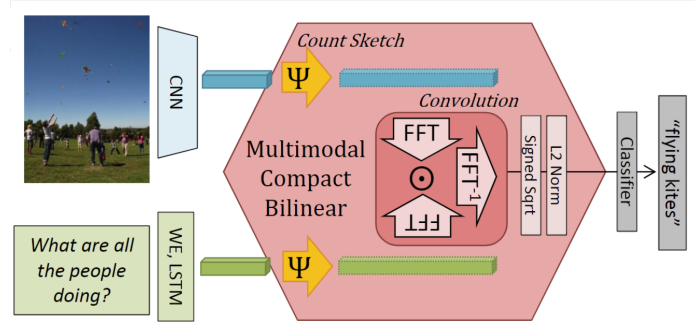


Figure 4.4: Bimodal bilinear pooling for visual question answering. Illustration taken from Fukui et al. [57]

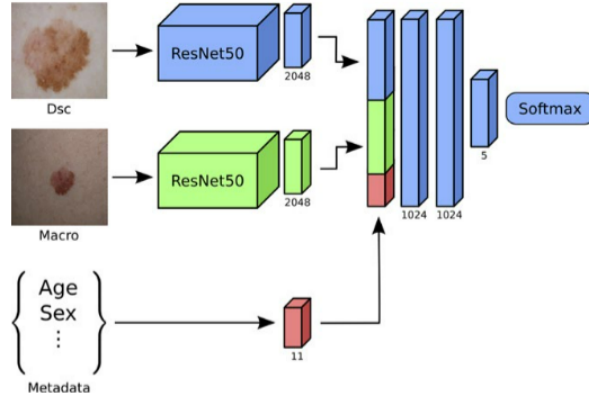


Figure 4.5: Diagram of the multimodal network for skin lesion classification. Schema taken form Yap et al. [58]

images of the concerned skin zone. For each of their patients, they had a binary histopathological diagnosis, where 0 is benign and 1 is cancerous. The first step was to extract the features from the images so that they can be fused with the metadata. Indeed, the raw image cannot be fused in early fusion with non-image data. To do this, they used the ResNet-50 architecture [59] along with transfer learning to obtain a good extraction of features. To represent the metadata in a learnable way, they performed a one-hot encoding, which converted them into binary vectors. To perform the fusion, they concatenated all the extracted feature into a single layer fed into a embedding network. Their approach is quite conventional when fusing images with metadata, but it works. Compared to the equivalent unimodal network, they performed 10% better in general. A schema of the network used can be seen in Figure 4.5.

An effective multimodal image fusion method using MRI and PET for the diagnosis of Alzheimer’s disease [60] In this example, multimodal imaging techniques are fused to generate a new image in which the target zone is emphasized. The goal is to create a network capable of performing an early diagnosis of Alzheimer’s disease by performing a prediction from medical images. To do this, MRI and PET were provided with the corresponding diagnosis. To perform the fusion, they first extracted the gray matter tissue from a brain MRI and used it as an anatomical mask. The extracted mask and PET scans are fused using a mapping operation. The advantage of this approach is that the resulting image, a GM-PET image, conveys a better description of the information than the individual input images, and as a single image, it can be used as input for any unimodal 3D CNN designed for this task.

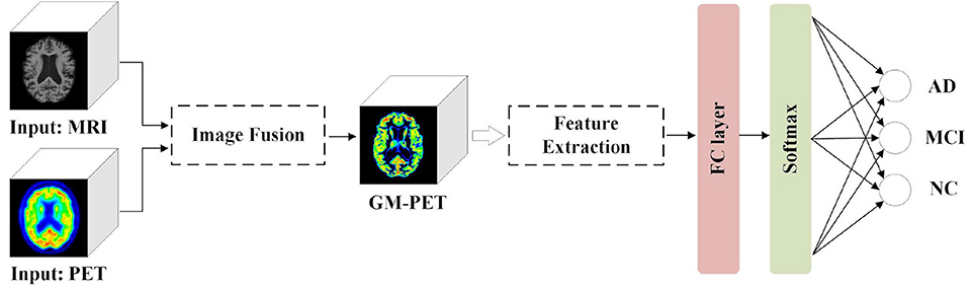


Figure 4.6: MRI and PET images fusion framework for early Alzheimer's disease diagnosis. Schema taken from Song et al. [60].

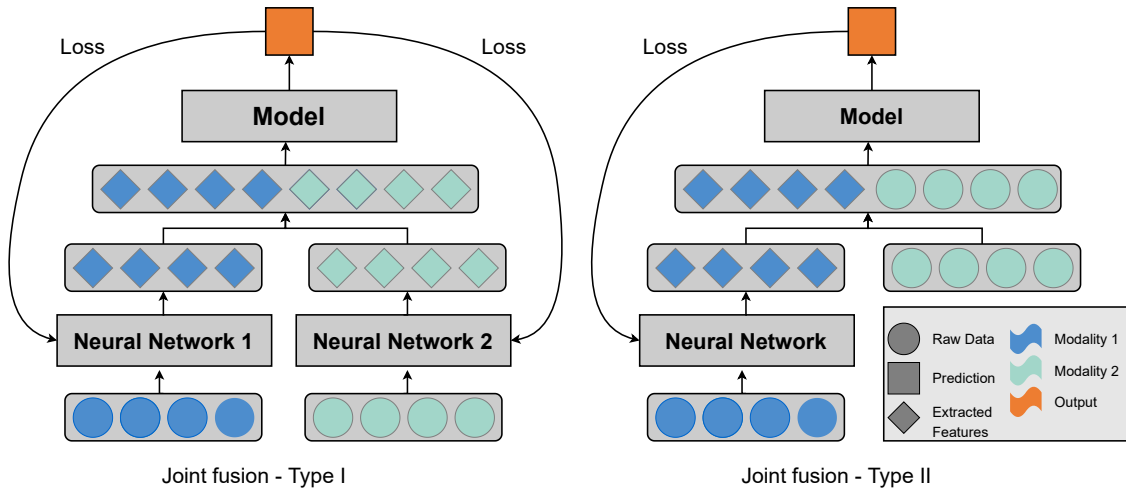


Figure 4.7: Example of architectures for type I joint fusion (the default intermediate fusion) and type II joint fusion (also named gradual fusion). Schema inspired by Huang et al. [53].

The implemented framework can be seen in Figure 4.6. Another important point is that the resulting GM-PET image can also be used by a practitioner because it emphasizes key characteristics. With no big surprises, the multimodal network performed better than its unimodal counterpart.

4.3 Joint fusion

The next fusion technique, joint or intermediate fusion, is the preferred approach to multimodal learning [23]. The idea is to first transform all raw data from various modalities into higher-level representations before fusing them. This can be seen in Figure 4.7. In deep learning, the result of the fusion is commonly called a shared representation layer [23]. As shown in Figure 4.7, the difference from type II early fusion is that the loss of the final model propagates to the networks that extract the features of each modality. This means that at each training iteration, the marginal representations will improve [53]. What it has in common with early fusion is that the model will learn the joint representation of the modalities, which allows learning cross-modality interactions. It should be noted that intermediate fusion is more suitable for deep learning architectures because fusion can be performed by connecting each modality to a shared layer [26]. Thus, it has been the approach chosen for the application presented at the end of this master thesis.

There is also a variant of joint fusion, called type II joint fusion, as shown in Figure 4.7. The idea is that, for some modalities, no features are extracted and, thus, the raw data can be directly fused. It is a way to fuse indicators, for example, since no feature needs to be extracted from them.

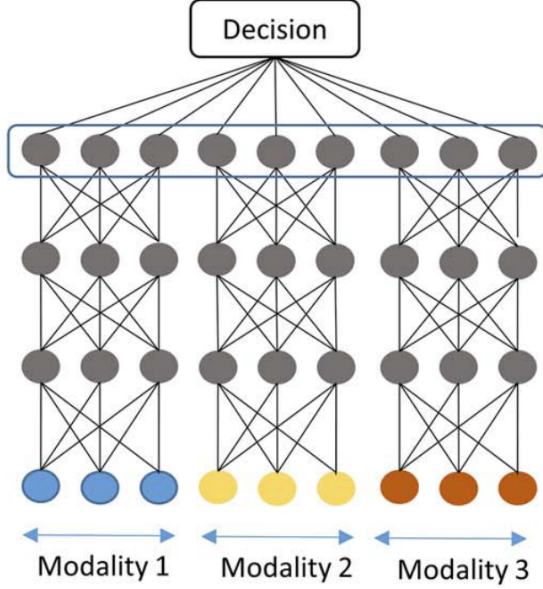


Figure 4.8: Concatenated vector directly input to the decision function. Illustration taken from Stahlschmidt et al. [26]

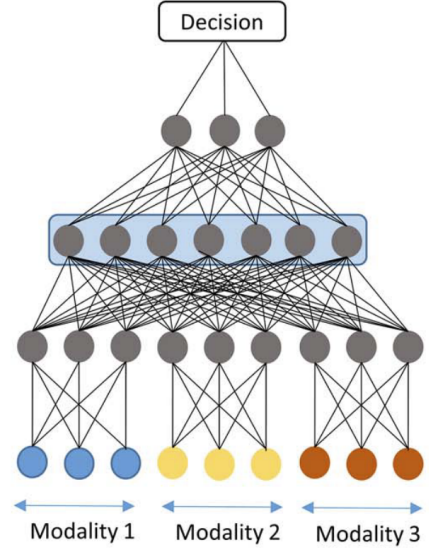


Figure 4.9: Concatenated vector used for joint representation learning. Illustration taken from Stahlschmidt et al. [26]

Intermediate fusion is interesting because it can combine the advantages of early and late fusion. During the first part, consisting in extracting the features for each modality, the different networks will learn marginal representations and will be able to discover within-modality interactions. After the fusion of the representations, which is usually a concatenation of the feature vector, two choices are possible.

In the first case, the decision function will directly take the concatenated vector as input, and we will speak of marginal fusion for this approach (left schema in Figure 4.8). In marginal fusion, we lose the advantage of learning the joint representation. However, the complexity of the model is lower and therefore reduces the risk of overfitting. Like late fusion, this approach can be interesting if the modalities used influence the result in a more independent way. In this case, the cooperative aspect of the modalities is less significant than complementarity and redundancy.

The other option, called joint fusion, is to continue the learning process to learn the joint representation of multiple modalities (right schema in Figure 4.9). Joint fusion makes the hypothesis that informative cross-modality interactions exist and do not independently affect the desired output of the network.

Another advantage of intermediate fusion is that it is possible to design a network that is robust against missing data or modalities. This is what Thung et al. [61] implemented with a network using biomedical data from multiple modalities to improve the diagnostic accuracy of neurodegenerative diseases. In the medical field, it is not rare that certain data are missing for some patient. This approach is in fact similar to co-learning, where a prediction task is assigned for each modality and the learning is based on the joint representation of these modalities. This approach provides supplementary information for the target task.

The main drawback with joint fusion is the complexity of implementing such a fusion. The network should be chosen and designed according to the fact that multiple modalities will be used. The main questions are when, how, and which modalities can be fused [23].

Another approach, almost unexplored, to joint fusion is gradual fusion [62]. Since the field of multimodal learning is quite broad and it is in its infancy, some papers will give different names for this

kind of fusion, but in this work, we will keep this terminology. The idea of gradual fusion is that highly correlated modalities fuse earlier, whereas less correlated modalities fuse later. This can be explained by the fact that correlation can be found more easily in more abstract representations.

In fact, the way feature vectors are fused plays a role in the sense that a simple concatenation may not be enough for the network to learn the cross-modality associations. Ramachandram et al. [23] also warned that naive fusion can lead to overfitting. To address this problem, performance can be improved by applying some dimensionality reduction techniques, but a better way to tackle this is to fuse feature representations in a more reflective manner, as seen in early fusion techniques.

When we try to solve a problem in machine learning, one of the big questions we want to know is which model is best for the given problem. The answer depends in part on the type of data with which the model will work. In multimodal learning, different types of data are used and, therefore, it can be interesting to use different models for each of them. Stahlschmidt et al. [26] proposed in their paper to distinguish between architectures that use only neural networks of the same type and those composed of different neural networks. They are respectively named homogeneous network design and heterogeneous network design. When we talk about the different networks used, we are talking about those used before the fusion to extract the features of the marginal representation.

4.3.1 Heterogeneous network design

As mentioned above, the main advantage of such an architecture is that, for each modality, the marginal representation will be better learned if these modalities are very different. The direct consequence of this is that, after fusion, the concatenated vector will better represent the higher-level features of the heterogeneous data.

For example, Venugopalan et al. [63] fused MRI (magnetic resonance imaging), genetic data, and clinical test data using different types of network. To extract the features from the MRI, a 3D-convolutional neural network was used, while for the genetic and clinical test data, they used stacked denoising auto-encoders in order to extract meaningful representation in the feature space.

4.3.2 Joint fusion in application

A multimodal deep learning-based drug repurposing approach for the treatment of COVID-19

[64] One of the approaches to combat the recent epidemic of COVID-19 is to find new therapeutic applications of existing drugs rather than develop newer ones. The study by Hooshmand et al. utilized a multimodal restricted Boltzman machine to combine two types of data from existing drugs, i.e. chemical structure data and differential gene expression in order to generate cluster defining new drugs that look like those given in input. First, two separate RBMs are used to extract the features of each modality and find the specific probability distribution of them. The two RBMs converge into one common RBM that is used to integrate the features discovered and identify the probability distribution of the combined data. Their approach can be seen in Figure 4.10. The results were in the form of 12 categories of clusters from which it was possible to extract new drug characteristics. They showed that the cluster generated from medicine developed specifically for COVID-19 matched with their cluster. Overall, their implementation outperformed the unimodal one by a 10% margin.

Combining structured and unstructured data for predictive models [65] The following example is also in the medical domain and aims to combine structured data such as electronic health records (EHRs) with unstructured data such as clinical notes to perform three predictions tasks. These tasks were in-hospital mortality prediction, long-term stay prediction, and hospital readmission prediction. Three data sets were available for this study: one with demographic information on the patients, one with EHR, and one last with unstructured sequential notes in the form of text information. To perform the fusion, they extracted the different features from each modalities and fused them into a high-level representation for each patient. Sequential notes were extracted using the Doc2Vec technique, a natural language processing technique. The features of the EHRs were extracted with a simple two-layer convolutional neural network, and finally the demographic data were simply encoded using a hot encoding technique.

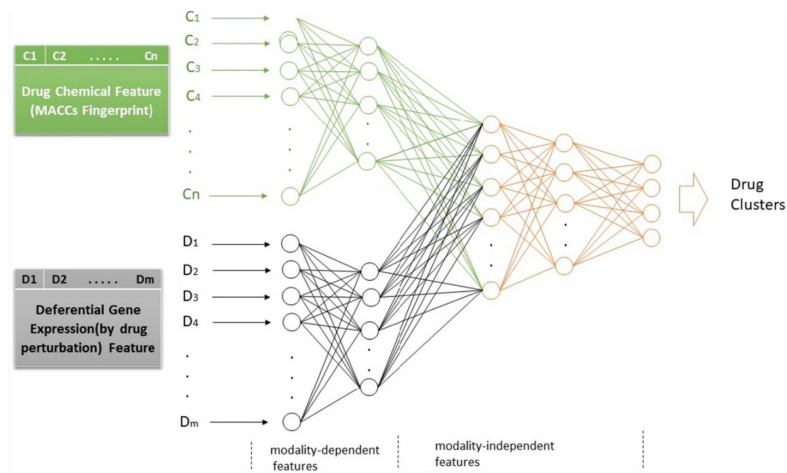


Figure 4.10: Illustration of the implemented multimodal restricted Boltzmann machine. Illustration taken from Hooshmand et al. [64].

The resulting joint representation is then fed into a neural network to make the final binary predictions. They showed that the more modalities were fused, the better the prediction was, showing that these modalities were bringing complementary information.

4.4 Late fusion

The final type of fusion is late fusion or decision-level fusion. The idea here is to aggregate the decisions or predictions of different models, each trained in different modalities, to obtain a new prediction. An example of a network architecture where late fusion is implemented can be seen in Figure 4.11.

Since each model is trained in a unimodal way, it allows us to learn the marginal representations, unlike the early fusion, which is mainly used to learn the joint representation. Using the same terms as in the section dedicated to early fusion, late fusion might ignore some low-level interactions between modalities [66]. One of the main reasons to choose late fusion over other approaches is that the errors of the different models tend to be uncorrelated. This implies that the different predictions have a complementary effect on the final prediction [23]. Another advantage of late fusion is that it makes it easier to deal with a situation where a modality is missing because the prediction will still be achievable [35].

The problem with late fusion is that it cannot learn the joint representation of the different modalities. Therefore, if the data from different sensors are correlated, late fusion may not be able to make the most of this fact. This is why Ramachandram and Taylor [23] recommend using late fusion when the input modalities are significantly uncorrelated, making the implementation much simpler. Another point of attention is the fact that each modality does not provide the same amount of information for the final prediction, and this fact should be taken into account when designing late fusion architectures [26].

However, the biggest question when working with late fusion is how to aggregate the predictions. The choice of the aggregation function, which is principally problem-dependent, may have a tremendous impact on the performance. The following part of the section will give a non-exhaustive list of aggregation functions and then will discuss different concrete applications of late fusion.

4.4.1 Aggregation functions

Choosing how to combine the results from different modalities is a key design choice and should be chosen in a way that takes advantage of the complementary information of each modality [26].

The most straightforward approach to combine the results of different models is to take the average of each individual output of the last layer of a neural network (which could be classes, probabilities, etc.),

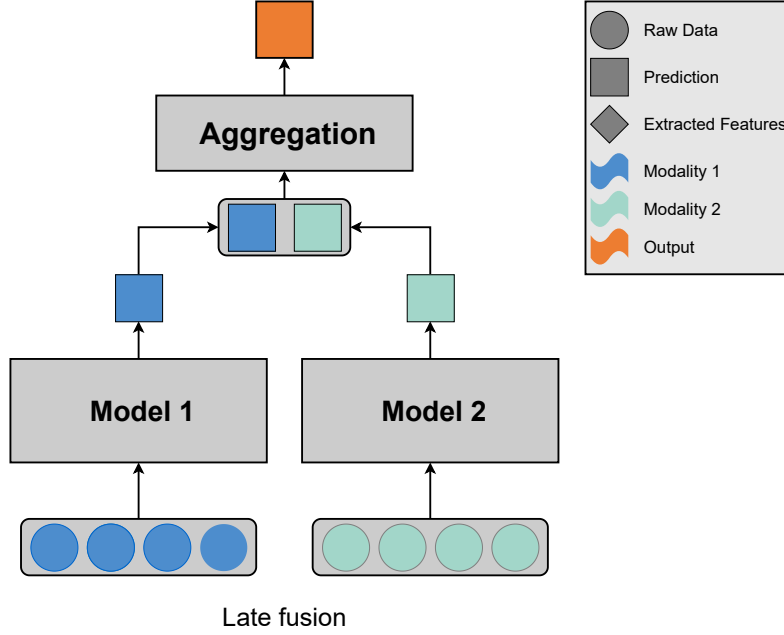


Figure 4.11: Schema inspired by Huang et al. [53].

but while it makes the model working, it is far from optimal in most situations. In fact, this approach assumes that each modality contributes the same to the final output.

Another way to aggregate the results is to use a voting scheme. In this situation, each model function is seen as a voter, and the goal is to find a weighted combination of them in order to minimize the misclassification rate while maximizing the voter's diversity. This can be done with a quadratic algorithm, such as MinCq [67], which would build a weighted majority vote while minimizing a risk bound [68]. Such a classifier can be written as

$$B_Q(\mathbf{x}) = \text{sgn}[E_{h \sim Q} h(\mathbf{x})], \quad (4.18)$$

where $B_Q(\mathbf{x})$ is the result of the weighted majority vote classifier B_Q , $h(\mathbf{x})$ represents the voters and $E_{h \sim Q}$ is the calculated weights.

These approaches are only two examples to give an idea to the reader how late fusion can be done, but in fact a very large number of aggregation functions exist. It will often depend on the nature of the task and the type of network(s) used to generate the multimodal predictions.

4.4.2 Late fusion in application

Deep learning-based late fusion of multimodal information for emotion classification of music video [66] In this example, Pandeya et al. addressed the issue of affective computing, an area of research that aims at training intelligent systems to recognize, feel, infer, and interpret human emotions. Music videos are particularly interesting for this task because it is possible to link the emotions coming from the music notes, the lyrics, and the visual expressions of the artist. The goal of this study will be to classify a music video into one of five emotion classes such as excitation, fear, or sadness. Each class is linked with several emotion adjectives: for example, sad is linked to anger, hate, and rage. To simplify a bit the architecture, they focused only on the audio and video part, dropping the lyrics. They trained separately two CNN for each modality, one 2D CNN for the audio input and one 3D CNN for the video input. Recall that the video signal has 3 dimensions. At the end of each network, a probability map is generated for each class. They concatenated these two maps and then applied a softmax function to recover the prediction. They showed that a multimodal architecture for this task boosted the overall

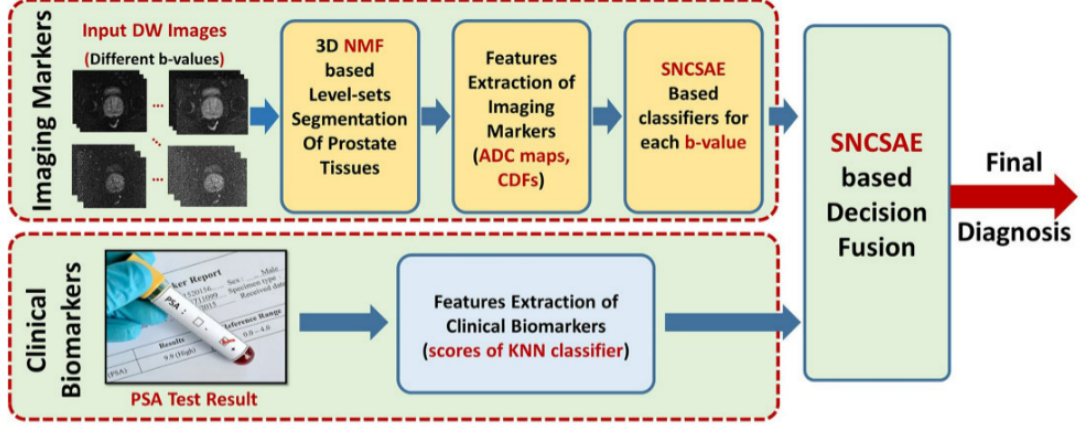


Figure 4.12: Schema of the framework implemented to perform a late fusion for prostate cancer diagnosis. Illustration taken from Reda et al. [69]

performance and reduced the rate of confusion. Another advantage they pointed out is the fact that combining information from multiple sources helped to overcome the lack of data problem.

Deep learning role in early diagnosis of prostate cancer [69] Reda et al. proposed in their article a computer-aided diagnostic system for the early diagnosis of prostate cancer. The goal was to use magnetic resonance imaging (MRI) and biomarkers to predict diagnosis. The framework implemented for this task can be seen in Figure 4.12. To make a prediction for the MRI, three steps were necessary. First, the CAD system performs a delineation of the prostate region using a level set-based model. Then, a feature extraction step is performed to extract apparent diffusion coefficient (ADC) maps that are then described using a cumulative distribution function (CDF). Finally, they used a stacked nonnegatively constrained sparse autoencoder (SNCSAE) to predict the diagnosis of the input prostate volume as benign or malignant in the form of a probability for each class. For the biomarker part, they used a k-nearest neighbor (KNN) algorithm to translate them into diagnostic probabilities. The probabilities of both networks are concatenated into a single probability vector $\mathbf{g} = [g_1, \dots, g_{16}]$ and fed into another SNCSAE. The SNCSAE determine the final classification of the prostate case as a probability for each output class c , using this formula:

$$p_t(c; \Theta_c^t) = \frac{\exp((\Theta_c^t)^T g_t)}{\sum_{c=1}^C \exp((\Theta_c^t)^T g_t)}; c = 1, 2 \quad (4.19)$$

where p_t is the probability that the diagnosis belongs to class c and Θ_c^T are the learned encoding coefficient for each class. The proposed approach outperformed the equivalent unimodal approach.

4.5 Discussion on Fusion Strategies

In this section, three types of fusion have been seen. Early fusion combines multiple modalities into a single vector or matrix to create a joint representation that will later be learned by the model. Joint fusion aims to first project the marginal representation of each modality into a high-level space before creating a joint representation in the same space. Finally, late fusion will be used to combine predictions from unimodal models, trained on marginal representation of each modality, to create a final multimodal prediction.

From the six examples presented here and from the literature, multimodal applications architectures outperform unimodal ones [26]. However, the results can also be influenced by the choice of the fusion,

which depends on the modalities used and the willingness of the researcher to make more or less architectural choices. It is still difficult to create a taxonomy between the three fusion strategies, since the performance of each remains task dependent. However, a general guideline would be that if the choice is between early and late fusion, early fusion could be more interesting for highly correlated datasets, while late fusion could be better with independent modalities [23]. Joint fusion combines the advantage of both early and late fusion but requires a more complex architecture that would highly depend of the task and the modalities, but may outperform by a good margin the two other fusion strategies.

The following section will show in detail how a joint fusion can leverage multiple modalities to increase performance on a small dataset by using complementary medical imaging.

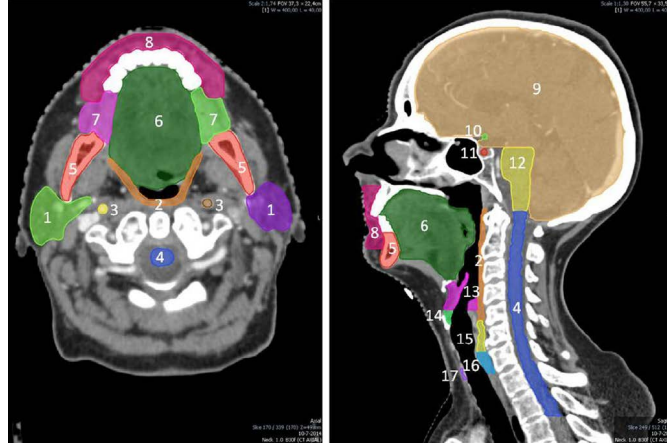


Figure 5.1: Illustration showing the segmentation of different organs in the head and neck region. Illustration taken from Liefbrig et al. [70].

5 X-Net: Multimodal Organs Co-segmentation

The second part of this thesis will cover an application that will serve as an example to prove that multimodal learning can be useful, particularly in the medical domain. The task is a segmentation of the gross tumor volume (GTV) in the head and neck of multiple patients with two imaging modalities.

This part starts with an explanation of the medical context and the motivations for performing the GTV segmentation. We will discuss what a GTV is, the two image modalities used to acquire patient scans, and the challenges of finding an automated solution to this task.

This will be followed by the objectives of this application from a deep learning point of view and, more specifically, from a multimodal deep learning perspective.

The next section will present different architectures used for this application. The U-Net will be introduced first since the following networks are based on it. This will serve as a base to introduce the X-Net, the core multimodal network for this application. Then the recently popular autoML (automatic machine learning) network nnU-Net and its multimodal counterpart, the nnX-Net, will finally be described.

Thereafter, a section will be devoted to the parameters used in both the U-Net and the X-Net. These parameters include some choices about the loss function, the optimizer, or also the batch size.

Finally, the results will be presented and discussed.

5.1 Medical Context and Motivations

Researches and innovations in oncology are of crucial importance, since cancer is one of the top five causes of death. Although cancer treatments are performed through several techniques, about half of them are treated with radiation therapy. Radiation therapy works like a canon that throws ionizing radiations at the target in order to kill the cells that compose it. Radiation therapy requires precision and therefore it is important to locate exactly the tumor region. This is done by performing an X-ray Computed Tomography (CT) scan and then conducting organ segmentation on it [70]. The problem with this approach is that the organ segmentation is done manually by an experimented practitioner and it is really time consuming. The segmentation of different organs can be seen in Figure 5.1.

The region of interest is the head and neck, leading to what is called head and neck cancer (HNC).

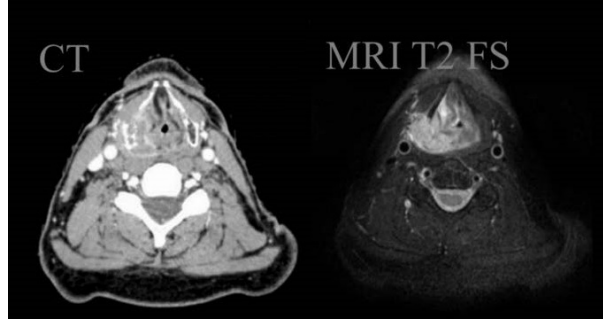


Figure 5.2: Images from the two modalities used for this task, created by selecting a precise slice in the z direction. Illustration taken from Liefgrig et al. [70].

The tumor can be found in several organs and structures of the head and neck, including, for example, the oral cavity or salivary glands, without mentioning them all. Interest in machine learning in medical applications grows each year and can potentially save medical practitioners a lot of time. In this application, the goal is to automatically perform gross tumor volume (GTV) segmentation. To do this, two types of medical imaging were used to obtain images of 20 patients. The first type is computed tomography (CT) and the second type is magnetic resonance imaging (MRI). The first is used for the planning of radiation therapy and the second is used for the diagnosis of the tumor.

For both modalities, the tumor volume has been given in a separate file and is called the label or the target. The images from both modalities and the labels are 3D matrices oriented so that the x and y dimensions represent the horizontal plane and the z the depth of the scan. A 2D image is obtainable by fixing one of the dimensions. Such thing is called slicing and the generated 2D image is called a slice. Images from both modalities can be seen in Figure 5.2.

More detailed information on the medical context can be found in the thesis of Éléonore Liefgrig [70] for the curious reader.

5.2 Objectives

Both modalities provide different types of information for the same region and therefore complementary information can be extracted from them. Therefore, the goal of this application will be to exploit and combine this information in order to perform a more precise tumor segmentation. At the same time, the goal is also to prove that a multimodal approach can outperform its unimodal equivalent, corroborating what was said earlier in this thesis. To do this, two networks will be trained: a unimodal U-Net and its multimodal counterpart, the X-Net. The results will be generated in the form of a score and will be compared to see if the multimodal approach makes sense. The results of Éléonore et al. [70] will also be used to have another reference for the benchmarking.

5.3 U-Net

The U-Net architecture was first introduced in 2015 by Ronneberger et al. [11] as a convolutional neural network for biomedical image segmentation, where, instead of classifying an image, each pixel receives a class label. The core motivation was that biomedical data sets tend to have only a limited number of annotated data, and hence these data should be used as efficiently as possible. To accomplish this, two components are used here. First, there is a strong use of data augmentation to increase the generalization capabilities of the network. The second component is the U-shaped network (see Figure 5.3) consisting of a contracting path (left side of the schema) and an expanding path (right side of the schema). The contracting path is used to capture the context, while the expanding path enables for a precise localization. Between them, there is what is called the bottleneck.

The contracting part acts like a conventional convolutional network and aims to extract the different

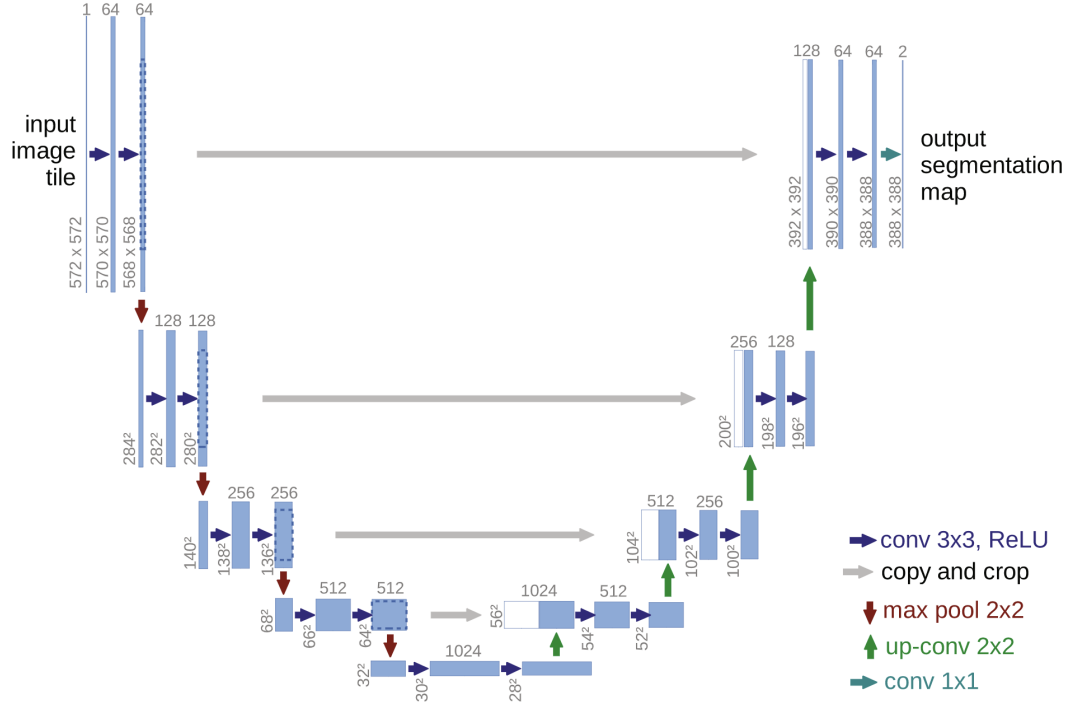


Figure 5.3: The U-Net architecture. Schema taken from Ronneberger et al. [11]

features from the input image and represent them at different levels of resolution. The contracting path is made up of 4 levels, each composed of 2 convolutional layers. Each of these convolutional layers is used with a rectified linear unit (ReLU), and each level is ended by a max pooling layer used for down-sampling. After each sampling step, the number of features doubles. The bottleneck is the same double convolution layer, but instead being followed by a max-pooling, it is followed by the expanding path.

The expanding path is more or less symmetric to the contracting path, which explains why the network is shaped like a 'U'. Each step starts with an upsampling of the output of the previous layer, followed by a convolution ("up-convolution"). This is represented by the green arrows in the schema. What is special in the U-Net is the skip connections, which are used to concatenate the corresponding cropped feature map from the contracting path. This allows the network to localize the upsampled feature maps and consequently, it allows successive convolutional layers to learn to generate a more precise output. Skip connections are gray arrows on the schema of the network. At the end of the expanding path, a final convolution is applied (light blue arrow) to have as many features as there are labels. If we look at 5.3, we can see that the segmentation map has a dimension of $388 \times 388 \times 2$, where 2 is the number of classes. While the input dimension of the images is 572×572 (in 2D), the output dimension will be smaller because at each convolution, the border pixels are lost (because the padding was omitted).

For this application, we used the 3D U-Net implemented by Mei et al. [71] which is shown in Figure 5.4. The source code was available directly on GitHub, and no modifications were made. Its contracting path is made up of five layers, represented by a red arrow, and each level is connected by a max-pooling layer, represented by a yellow arrow. As for the U-Net of reference, each layer is composed of two times the same block: a three-dimensional convolutional layer with a $3 \times 3 \times 3$ kernel followed by a batch normalization layer and then ended by a LeakyReLU activation function. The batch normalization layer is used to accelerate training by normalizing each layer of the network, to put it simply. The curious reader is invited to refer to the paper dedicated to this topic [72]. The LeakyReLU activation function is a ReLU function, but with a small slope for negative values (instead of no slope at all). We can write its equation as follows:

been used for aligned images and have surpassed their unimodal equivalent. For example, Jin et al. [73] performed early and late fusion for esophageal GTV segmentation with CT and PET imaging modalities. Although the early fusion model was already obtaining better results than the unimodal 3D U-Net, they added a late fusion to their model, again improving the results by a slight margin. They concluded that the proper use of complementary data was beneficial for the desired task. Yang et al. [74] even used three modalities (MRI, CT, and PET) for the same task as in this thesis. After registering PET and MR images in relation to CT images, they performed an early fusion, again leading to better results than the unimodal equivalent. All this greatly motivates the use of multiple modalities in medical segmentation.

5.5 nnU-Net

The U-Net presented earlier is a really good architecture, often used as a benchmark in the literature. The only problem with it is the choice of the different parameters depending on the task. In fact, the degree of freedom regarding these choices can become more or less pretty large. Isensee et al. [75] addressed this difficulty by presenting a self-adapting network, called nnU-Net ("no-new-Net"), capable of finding the right parameters depending on the task. This approach is commonly referred to as AutoML, which stands for automatic machine learning. While this approach is really convenient because it saves a lot of time that is used for fine tuning, it is also performing very well. This is particularly useful when the U-Net is used as a reference and not as the main subject of a study. At the time of writing this thesis, it was considered a state-of-the-art biomedical image segmentation tool [70].

Automatic network configuration is based on three main ingredients [76]:

1. **Fixed parameters:** decisions design that do not requires adaptation between the data set and the common configuration for them
2. **Rules-based parameters:** formulations of explicit dependencies between the specific properties of the data set and the design choices for as many of the remaining decisions as possible
3. **Empirical parameters:** learning the remaining decisions empirically from the data

The use of nnU-Net applied to the same task as in this thesis was the main subject of Éléonore's thesis [70]. This means that the different results obtained on the MR images are also valid and can be used as a benchmark for our X-Net. However, no CT predictions were made, so the U-Net will serve as a benchmark for this modality, which makes total sense when the goal is to prove that a multimodal approach performs better than the unimodal equivalent.

5.5.1 nnX-Net

As mentioned above, Éléonore has introduced nnX-Net and its design is almost the same as the X-Net presented in this thesis. The only difference is the absence of a supplementary fusion after the concatenation of the features at the bottleneck and the fact that it is based on the nnU-Net instead of the U-Net. This approach combines all the benefits of the nnU-Net and those of the X-Net. Although a large part of the time devoted to this thesis was used to debug the unfinished nnX-Net, it was not possible to achieve it. The main reasons are that the code is really complicated, and, due to the AutoML part, a small change in a function can have some big repercussions. Because there was a time delay at the end of the semester, attention has been paid to a simpler implementation, X-Net. Although the results expected by the X-Net are lower than the ones expected by the nnX-Net, the X-Net serves as a proof of concept for this architecture and motivates its implementation.

5.6 Methodology

This section presents how the dataset has been used for training and testing and what parameters have been used for network training. The first part presents the preprocessing of the images, the different splits of the datasets and, finally, the data augmentation steps to increase the generalization capabilities of the network. The second part describes in detail the parameters used for network training.

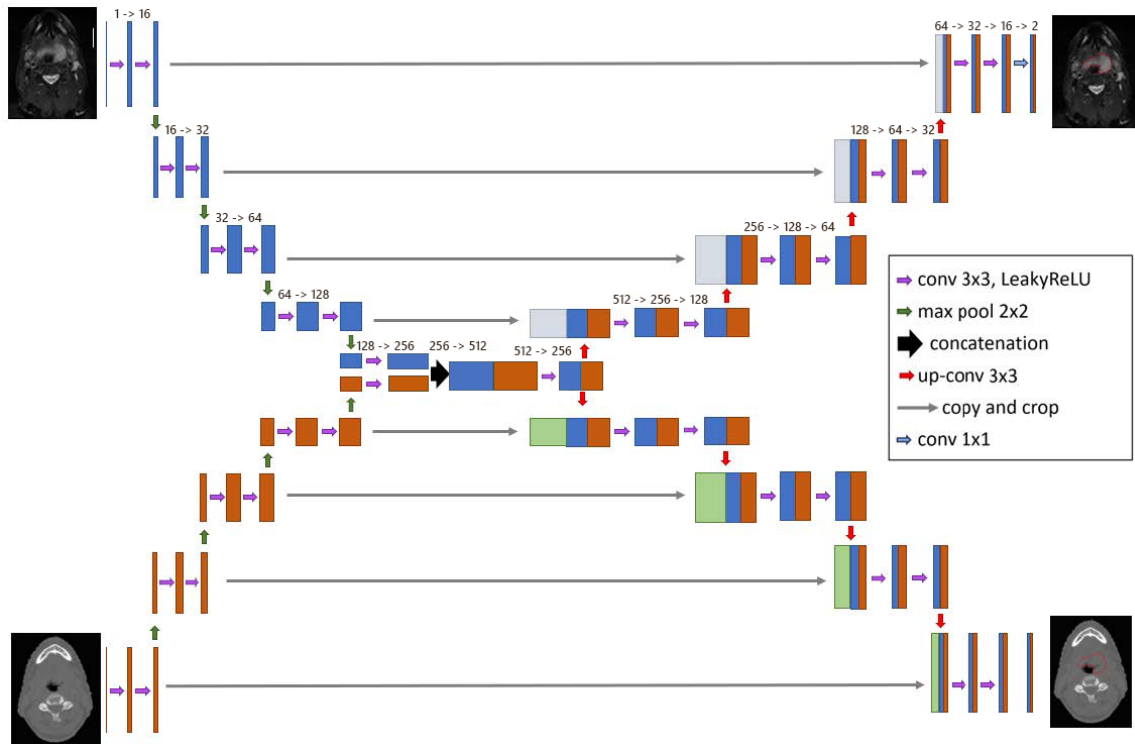


Figure 5.5: Illustration of the X-Net architecture. Each rectangular represent a feature map for the image and above each of them is indicated the total number of features. The color blue is associated with the MR images and the orange is for the CT images. Schema adapted from Lieffrig et al. [70]

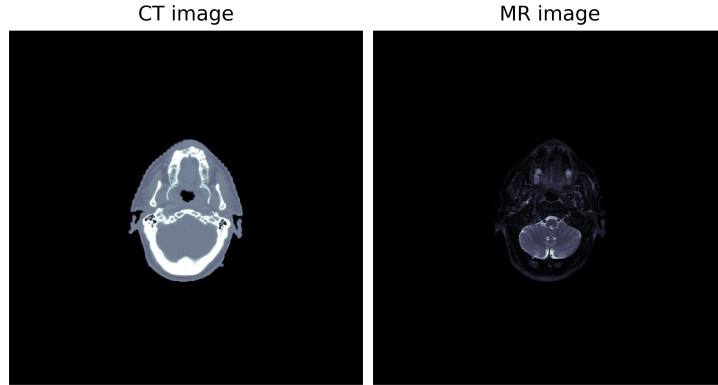


Figure 5.6: Example of what the images look like for the two types of modalities.

5.6.1 Preprocessing and data augmentation

In this section, different information is presented relative to the data used here, as well as the modification that has been made to it.

Datasets description The two data sets are made up of 20 patients named ANONXXXX, the Xs represent a number to uniquely identify each patient anonymously. For each of them, a 3D image from each modality (MR and CT) was available. For both modalities, one 2D slice has a resolution of 512 by 512 voxels, and the third dimension varies between patients but not between modalities, and its mean value is 142 slices. For each patient and for each modality, a corresponding label has been assigned, they have exactly the same dimensions as the corresponding images, and it is represented by a Boolean matrix. The voxels corresponding to the background in the label are equal to 'False' while the voxels of the GTV are equal to 'True'. The value of the voxels in the raw CT scans are on average between -16768 and 1560 and for the MR images they are between 0 et 1740. An important point is that the CT and MR images are not totally aligned since two weeks separate their acquisition. However, a program has been used to perform rigid registration to obtain more aligned images. An example of the data can be seen in Figure 5.6.

Preprocessing The first step of the preprocessing is to convert the data into NIfTI (Neuroimaging Informatics Technology Initiative) images. The raw data are in the form of a compressed numpy array. First, the intensities of the CT scan are cropped to fit in the interval $[-1000, 1000]$: all values lower than -1000 are put at -1000 and all values superior to 1000 are put at 1000 . In doing so, we arrive at a uniform background around the body worth -1000 . Second, the labels are converted into binary arrays by replacing 'False' entries with 0 and 'True' entries with 1. Finally, the arrays are saved in NIfTI format where the affine function is the unit matrix with four dimensions. The affine function gives information on the orientation of the patients in a reference and the size of the voxels (in mm).

The second part of the preprocessing begins with cropping. In order to work only with images having the same dimension, a common crop zone is computed and then applied to all images and labels. First, for each image, a cropping zone is computed. The goal is a smaller zone to work with so that we can get rid of the non-body voxels. An example can be seen in Figure 5.7. A common cropping zone is computed by taking the smallest zone including all the cropping zone. This zone is adjusted so the final size of the images in directions x and y are divisible by 16, because of network constraint. Another step consists in making a final cropping of the intensities of the CT images to work only in the range $[-400, 400]$, where all the useful information on soft tissues and consequently the tumor, lies. Finally, for both CT and MR images, the intensities are normalized between 0 and 1. Once the preprocessing has been done, the data is uploaded to the external server where the computations will be performed.

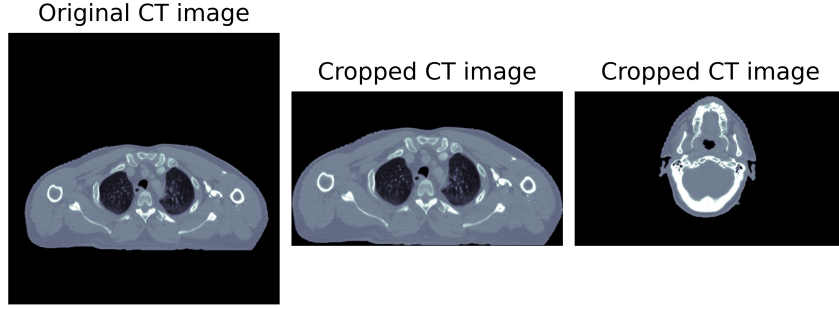


Figure 5.7: Illustration of the cropping applied individually to a CT image. The two left images represent the same slice while the right image represent another slice. This shows that the cropping has been made such that no voxels in the body is lost while minimizing the number of background voxels.

Data split In order to generate a prediction for each of the 20 patients, a 5-fold test has been used. This means that the network will be trained five times, each time on a different data set. For each training, 16 patients have been used for network training and four for testing. If we take for each time four different patients for the testing, we arrive at the expected twenty predictions (for the X-Net, it will be forty since there are two modalities). It is worth noting that the splitting has been made randomly.

Data augmentation To keep the code relatively simple, only two data augmentation techniques have been used. The first augmentation is a random cropping which just randomly cropped the image. The second data augmentation is a random flip of the image that randomly rotates the image by a certain angle.

5.6.2 Models parameters

This section presents the different parameters used for the training of both the U-Net and the X-Net. These two networks share the same parameters to have a meaningful comparison between them.

For this task, the Dice loss function has been chosen with an Adam optimizer. The motivation of using the Dice loss is the fact that for some patient (and in general in medical segmentation tasks), the region of interest, i.e. the tumor in our case, can be very small compared to size of the scan, which can causes to be stuck in local minima during the learning process. The Dice loss correct this by balancing the importance given to the region of interest and the background. This approach is based on the derivative of the Dice coefficient (presented in the next section), given by:

$$\frac{\partial D}{\partial p_j} = 2 \left[\frac{g_j(\sum_i^N p_i^2 + \sum_i^N g_i^2) - 2p_j(\sum_i^N p_i g_i)}{(\sum_i^N p_i^2 + \sum_i^N g_i^2)} \right] \quad (5.21)$$

where D is the Dice coefficient, p_j and g_j respectively the predicted binary segmentation and the ground truth for voxel j and N the total number of voxels [77].

Working with 3D images makes the number of parameters to update very large. The Adam optimizer has been created to deal with a large number of parameters while having a little memory requirement. This approach is based on stochastic optimization and has demonstrated excellent results with convolutional neural networks [78]. Three parameters are linked to this optimizer: the learning rate, the momentum, and the weight decay. The learning rate has been explained at the beginning of this thesis and for our networks the choice has been 10^{-4} . The momentum is a technique used to reduce the impact of noises in convergence to the optimal weight by giving to the optimizer a more stable direction gradient. The value of 0.9 is typical for deep learning applications. The weight decay is a parameter used in the algorithm of the Adam optimizer and its value of 10^{-5} is a default value.

Parameter	Value
Activation function	LeakyReLu
Negative slope	0.01
Batch size	16
Loss function	Dice Loss
Optimizer	Adam
Learning rate	10^{-4}
Momentum	0.9
Weight decay	10^{-5}
Max iterations	17000
Device	GPU

Table 5.1: Training parameters for U-Net and X-Net

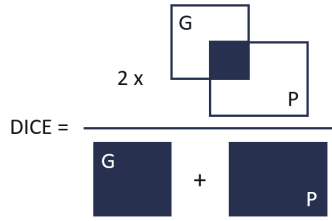


Figure 5.8: Dice coefficient computation in illustrative way. Schema taken from Lieffrig et al. [70]

The last piece was to train the networks on a GPU, in our case a nVidia A100 with 40Gb of HBM2 memory. The access to this GPU has been granted by the Consortium des Équipements de Calcul Intensif (CECI).

A summary of the training parameters can be found in Table 5.1.

5.6.3 Testing

While there exist many different ways to assess the performance of a network for a segmentation task, here only one metric has been used, since the only objective of this application is to make a comparison between networks. The metric used here is the Dice coefficient, which is probably the most widely used for this kind of task. It is defined as follows:

$$D = \frac{2|G \cap P|}{|G| + |P|} = \frac{2TP}{2TP + FP + FN} \quad (5.22)$$

where G is the point set containing the ground truth of the segmentation and P the point set containing the predictions. TP stands for true positive meaning that a voxel of the target volume has been correctly labelled as a GTV voxel. FP stand for false positive, meaning this time a background voxel received the label of a GTV voxel. Finally, FN stands for false negative, meaning that a GTV voxel has been incorrectly labeled as a background pixel.

The Dice coefficient is interesting because it is not just about checking for each voxel if the prediction is correct. In this case, there would be a really large class imbalance since there are a lot of background voxels and far fewer target voxels. The dice coefficient allows instead a direct comparison of the ground truth and the predictions. However, the Dice coefficient has some limits, as with very small target regions. In this case, only a few voxels difference between prediction and ground truth can lead to a huge drop in the Dice coefficient [70]. An illustration of the Dice coefficient can be seen in Figure 5.8.

Architecture	U-Net	X-Net		nnU-Net
Modality	CT	CT	MR	MR
ANON242	0.08	0.24	0.55	0.51
ANON590	0.28	0.48	0.6	0.85
ANON690	0.35	0.45	0.52	0.83
ANON866	0.15	0.52	0.46	0.53
ANON1132	0.53	0.56	0.54	0.4
ANON1677	0.61	0.58	0.65	0.78
ANON1783	0.32	0.44	0.11	0.66
ANON2453	0.2	0.02	0.14	0.31
ANON2834	0.29	0.31	0.75	0.69
ANON2865	0.49	0.68	0.81	0.71
ANON2971	0.06	0	0.2	0.46
ANON3146	0	0	0	0.21
ANON3207	0.26	0.34	0.69	0.85
ANON3435	0.42	0.61	0.58	0.81
ANON3620	0.3	0.14	0.7	0.77
ANON3647	0.12	0.35	0.51	0.56
ANON3648	0.25	0.31	0.44	0.16
ANON4232	0.63	0.66	0.75	0.84
ANON4297	0.46	0.43	0.26	0.64
ANON4641	0.28	0.37	0.68	0.58
Overall	0.304±0.18	0.3745±0.21	0.52±0.25	0.61±0.21

Table 5.2: Results obtained for the three architectures U-Net, X-Net and nnX-Net (Liefgrig et al. [70]). The values are the Dice scores computed for each patient, for each architecture, and for each modality used with these architectures. Two comparisons are made: one between the U-Net and the X-Net for the CT images and one between the X-Net and the nnU-Net for the MR images. For each comparison, the best values are in bold. The last row of this table shows the mean value and standard deviation of all patients.

5.7 Results and Discussion

In this section, the results of the two architectures studied for this application, i.e. the U-Net and the X-Net, are presented and analyzed. The results obtained with the nnU-Net by E. Liefgrig [70] for magnetic resonance imaging modality will be used to give an idea of the performance of the X-Net compared to a reference model. The Dice scores obtained are shown in Table 5.2 and all the results for the U-Net and X-Net can be found in the appendix.

5.7.1 General observations

X-Net The results obtained with the new multimodal X-Net architecture are quite good. In general, the network was able to produce better predictions for MR images than for CT images. This can be explained by the fact that the tumor area is easier to discern on MR images than on CT images, because the soft tissues are more contrasted. It also appears that there are large differences between the results of the patients. This point will be discussed in more detail in the next section.

We can also observe that, for some patients, the X-Net performed as well as, or even better than, the nnU-Net. However, this observation should be taken with caution because the training and testing datasets of the two architectures (5-fold testing) are not the same, and with 20 patients this can have

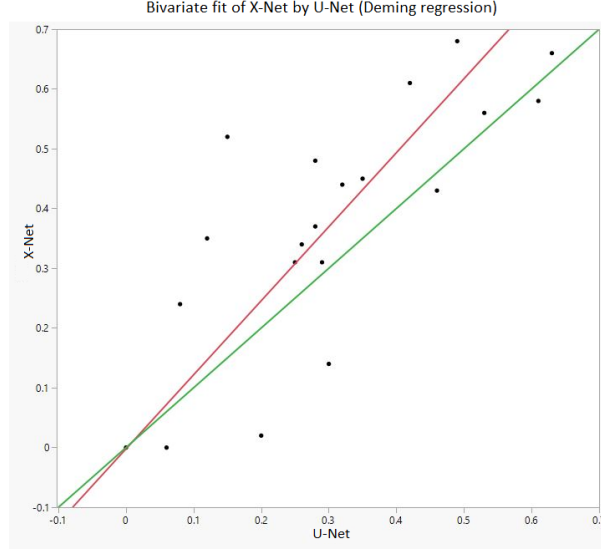


Figure 5.9: Bivariate fit with a Deming regression [79] of X-Net results by U-Net results where the red line is the the regression and the green line is the unitary line $x = y$ used as reference for the comparison.

an impact on the results. However, these results are encouraging when we recall that nnU-Net is a top performer in the field of organ segmentation.

U-Net and X-Net We can see in Table 5.2 that the mean results (with standard deviation) for U-Net and X-Net on CT images are respectively 0.304 ± 0.18 and 0.3745 ± 0.21 and that for 15 patients on 20, X-Net performed better than U-Net. To properly compare U-Net and X-Net, a Deming (or error-in-variable) regression [79] has been performed since we have two observations with errors. The equation of this linear regression can be written as follows:

$$y = \text{intercept} + \text{slope} * x \quad (5.23)$$

where x represents the U-Net results and y represents the X-Net results. The intercept and slope values are, respectively, ~ 0 and 1.238 and the bivariate fit resulting from this regression can be seen in Figure 5.9. The values of these parameters can be interpreted as "X-Net results are estimated to be ~ 1.24 times better than U-Net results". However, with a significance level of 0.05, X-Net is not statistically better than U-Net. With this level of significance, the computed confidence interval is $[0.827, 1.929]$, where our slope value lies. Although this seems to contradict our intuition, it can be explained by two facts. First, the dice scores are in the interval $[0, 0.68]$, and second, the sample size is only 20 patients, which means that our data points are widely scattered and therefore it is difficult to draw a conclusion from them. A bigger data set should bring more diversity and should probably show that X-Net is indeed better than U-Net. However, taking into account the different works that have been done in the field of organ segmentation, a gain of 24 percent on average is quite important; hence, the need for a larger data set.

Although we have not reached statistical difference, these results can serve as a proof of concept for the multimodal approach of X-Net in the sense that it can use two modalities to increase the quality of the predictions. Now, it would be interesting to test it not only with larger data sets, but also with different modalities and for different medical segmentation tasks, since the network could work with other types of data with only minor modifications. These results are also a strong motivation to continue working on the nnX-Net architecture. If the nnX-Net can combine the advantages of the X-Net with the advantages of the nnU-Net, this network could be one of the in this domain when multimodal data are available.

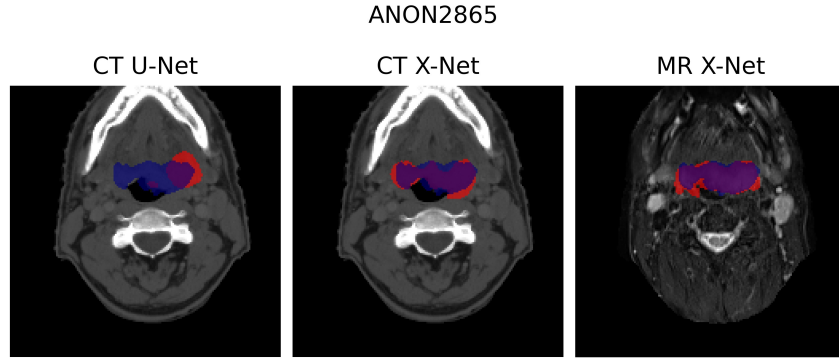


Figure 5.10: Ground truth (blue) and predicted GTV (red) masks for patient ANON2865. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.49, 0.68 (CT) and 0.81 (MR).

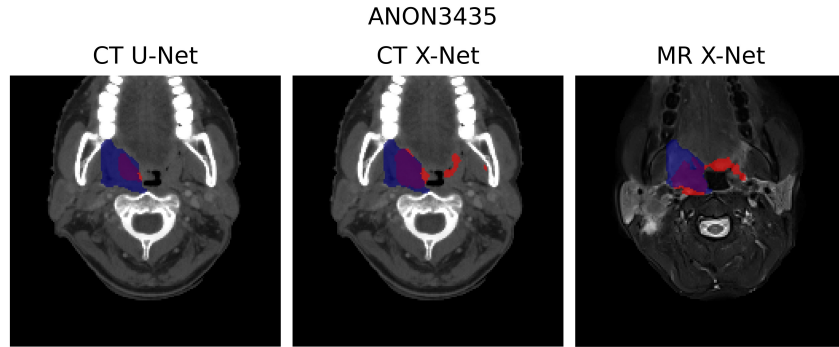


Figure 5.11: Ground truth (blue) and predicted GTV (red) masks for patient ANON3435. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.42, 0.61 (CT) and 0.58 (MR).

5.7.2 Special cases analysis

In this section, we are interested in a couple of special cases for which the prediction has been particularly good/bad or for which there is a big difference between the two architectures of interest.

Best 3 results As shown in Table 5.2, the best predictions obtained with X-Net are for patients ANON2865 (Figure 5.10), ANON3435 (Figure 5.11) and ANON4232 (Figure 5.12). The gross tumor volume (GTV) for these patients is quite large and has no particular shape, which can explain why they have the best scores. For ANON2865 (Fig. 5.10), we can see that the addition of the MR image provided additional information on the target location because the U-Net completely missed the left part of the GTV. It seems that is also the case for the two other patients; however, for patient ANON4232 (Fig. 5.12), it seems that the U-Net was already performing well but with the added modality, the network was able to slightly shift the GTV toward a better segmentation.

Worst 3 results The three worst results have been obtained for the patients ANON2971 (Figure 5.13), ANON2453 (Figure 5.14) and ANON3146 (Figure 5.15). The prediction for ANON2991 is completely out of place. We can observe that the GTV volume is really big and it seems that here there is not much contrast with the background. Another possibility is the fact that this kind of very large GTV is not well represented in our data set, meaning that the network is in an unknown terrain. The big surprise with this patient is the fact that it has a prediction for the MR image but not for the CT image, where

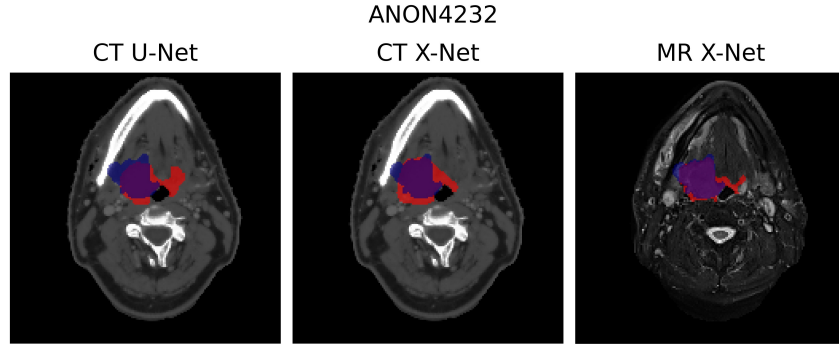


Figure 5.12: Ground truth (blue) and predicted GTV (red) masks for patient ANON4232. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.63, 0.66 (CT) and 0.75 (MR).

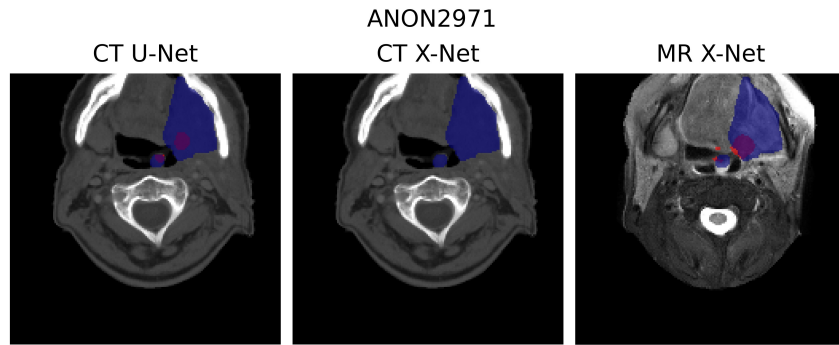


Figure 5.13: Ground truth (blue) and predicted GTV (red) masks for patient ANON2971. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.06, 0.0 (CT) and 0.2 (MR).

the U-Net has at least something for the CT images. An hypothesis would be that because the fusion in the X-Net has been done before any skip-connections, no localization information is shared and maybe performing the fusion later in the expanding path can help. For patient ANON2453 (Fig. 5.14), the GTV is very small and there is a really large inflamed nodule that could have triggered the network. The nodule is much more contrasted in the MR image and could influence the results of the X-Net. The fact that the U-Net does not appear to have been influenced by the nodule confirms that the MR image negatively influenced the results. Finally, for patient ANON3146 (Fig. 5.15), no segmentation was performed. Looking at Table 5.2, we can see that nnU-Net also did not perform well for this patient. Regarding the explanation for patient ANON2971, this GTV should also be an outlier in the data set, and therefore the network was unable to recognize any familiar sign of GTV.

Other special case Patient ANON3620 figure 5.16 is interesting because the prediction of X-Net for the CT image is worse than the prediction of U-Net and, at the same time, the prediction for the MR image is really good. In some cases, perhaps the fusion can have a destructive effect on the representation of the GTV in the network. But this does not answer why the network performed so well on the MR image. As mentioned earlier, it might be a good idea to try to perform the fusion later in the expanding path. Another option would be to have a kind of late fusion that takes into account the probability for each segmentation (of each modality) and adapts the prediction consequently. If the probability is high for the MR GTV prediction and not low but not high either for the CT GTV prediction, maybe the MR probabilities could "pull up" these probabilities. Even if these two modalities are not perfectly aligned, the GTV is still pretty much in the same location, so it could potentially help.

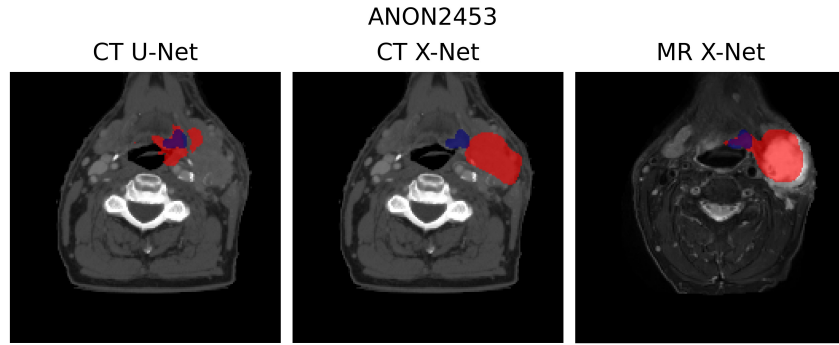


Figure 5.14: Ground truth (blue) and predicted GTV (red) masks for patient ANON2453. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.20, 0.02 (CT) and 0.14 (MR).

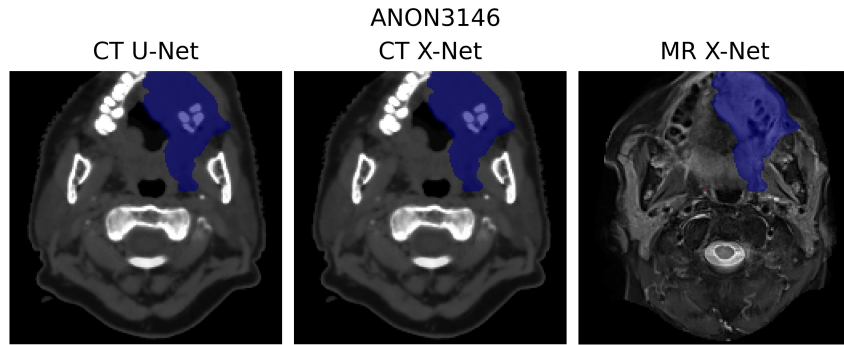


Figure 5.15: Ground truth (blue) and predicted GTV (red) masks for patient ANON3146. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.0, 0.0 (CT) and 0.0 (MR).

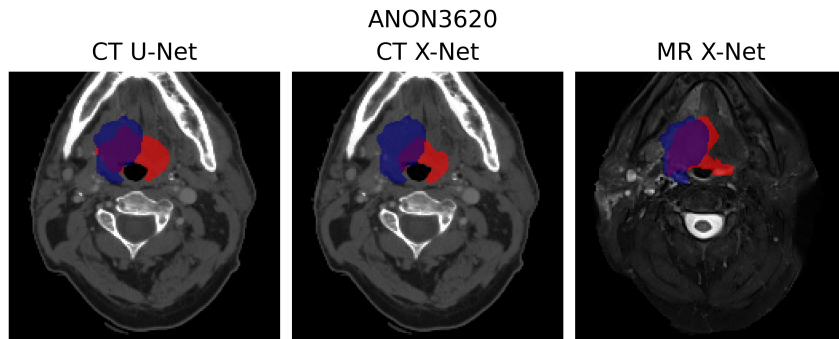


Figure 5.16: Ground truth (blue) and predicted GTV (red) masks for patient ANON3620. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.3, 0.14 (CT) and 0.7 (MR).

5.7.3 Discussion

While the results were particularly good, it is important to recall that the modifications made to the U-Net to obtain the X-Net were minor. This means that there is much room to fine-tune the architecture and the parameters. For example, only two data augmentation techniques were used, and one can add, for example, random elastic deformations, gamma correction augmentation, or also mirroring [75]. Since the nnU-Net has already many augmentation functions implemented, this is another motivation to continue working on this architecture. The advantage of data augmentation, especially for small data sets, is that it increases the generalization capabilities of the network. As we saw, some patients are not well represented in the data set and adding more data augmentation techniques may help. Other loss functions can also be used as focal loss, for example, which Tang et al. [80] used with their multimodal architecture for glioma segmentation. In segmentation task, cross entropy and weighted cross entropy loss have also been used.

Another point that could help is to add some post-processing steps. One thing that can be done is to remove non-primary GTV. For some patients, it has been possible to observe that the network predicted multiple GTVs, which in this case cannot happen. It could be done by adding a function that identifies the main GTV (biggest one or best probabilities) and then keeps only the voxels belonging to it. Also, as discussed earlier, we could implement a kind of late fusion taking the prediction probabilities of both modalities in order to have a sort of transfer between them. As we saw with ANON3620, if one modality performs better than the other, it could compensate for that. This kind of fusion has been implemented by Tang et al. [80] using a fusion network for the two outputs. The main difference from their approach is that our modalities are not aligned. Maybe it would be possible to keep only the probabilities for voxels being at a certain distance from the border of the GTV because the center of it should be the same for both modalities. It would also be interesting to make a comparison between the architecture of Tang et al. [80] and the X-Net to see which performs better even on non-aligned modalities.

Something that was missing in the results was MR images predictions from the U-Net. Due to the lack of time and the long training time (approximately 24 hours), it was not possible to obtain results for this modality. However, it could be really interesting to see how the X-Net performs against the U-Net for the MR modality. This can be done in about one week, so it is definitely worth trying. The network could also be trained with other data sets, which could give more results to compare the U-Net with the X-Net. Not only larger datasets, but also other combination of modalities such as MRI + PET or CT + PET. The network should work fine, even if we change the modalities.

The place where the fusion is performed in the network may also be a factor in the quality of the joint representation of both modalities. Future work must be done to understand the impact of moving the fusion inside the expanding path. An hypothesis would be that the more the images are aligned, the closer of the output the fusion can be. In a certain way, it could replace the late fusion described earlier.

The final point is the fact that the presence of other cancerous cells can influence segmentation. This is the case for the large nodule seen in ANON2453 (Fig. 5.14). The volume of cancerous nodules (GTVn) should normally be located in different places than the GTV. This information could be exploited by an attention network as implemented by Mei et al. [71].

6 Perspectives

In this section, several tracks for future work are given regarding what has been said in this thesis. For the application (X-Net), this information has already been given in the discussion of the results.

- **Model-based fusion:** the vast majority of the work done on multimodal fusion involves modifying unimodal architectures to add new modalities. Model-based fusion involves building machine learning models that are designed from the ground to work with multimodal data, and more research should be done in that field.
- **Co-learning:** Co-learning is also a promising way to use multimodal data, especially when more modalities are available during training than during inference. This is also motivated by the fact that multimodal data become more available.
- **Transfer learning for multimodal applications:** transfer learning is commonly used to transfer pretrained weights from a certain task to a network used for a similar task. It could be interesting to see how transfer learning works when weights have been trained in a different modality.
- **Modality selection:** while it is very interesting to use as many modalities as possible, adding some modalities can have a detrimental effect or no added benefit. Although feature selection exists already for homogeneous data sets, the comparison of modalities is quite different, and more research should be done in this field.

7 Conclusion

This thesis has been divided into two parts. The first part addressed the question of the use of multimodal data in machine learning in a theoretical way. The second part has approached this problem in a practical way with an application example.

From the theoretical part of this thesis, it is important to remember that there are five main challenges in multimodal machine learning. First, fusion, which has been studied in more detail, where multimodal data are combined in order to train a model. Second, multimodal representation was discussed, in which the aim was to find the best ways to represent multimodal data. The multimodal translation followed, where it was a question of understanding how a modality can be expressed in the space of another. Multimodal alignment is about how to match data points from one modality with data points from another. Finally, co-learning has been discussed and is interested in the question of how the addition of a modality can allow a better understanding of the information present in the base modality.

The application part of this thesis was dedicated to X-Net, a multimodal architecture adapted from U-Net to take advantage of the complementarity of CT and MRI images. The results served as a proof-of-concept for this architecture, and many possibilities for future work were explored. This architecture is still in its infancy, and more tests and refinements are needed. This includes:

- Analyze the impact of the position of the fusion in the expanding path
- Train the network on larger data set
- Perform a post-processing to clean the segmentation
- Train the U-Net on the MR images
- Perform a probabilities fusion with the predictions

The results from X-Net serve also as motivation for the enhanced version of it, nnX-Net. Making this architecture work is more complicated than it appears, but the potential is there. It could be a good idea, based on the implementation of X-Net, to reconsider how to implement nnX-Net. The AutoML part should be more of an ally than the bad guy that makes every little modification full of consequences.

To conclude, heterogeneous data in machine learning offer a lot of opportunities and have shown in past applications how powerful it can be. This has been shown through numerous examples that have combined all kinds of modalities. It was also seen that the use of multimodal data is not specific to one domain, but that, on the contrary, all domains benefit from it, whether it is in the medical field, in video transcription, or even in the development of drugs.

A Results U-Net and X-Net

ANON242

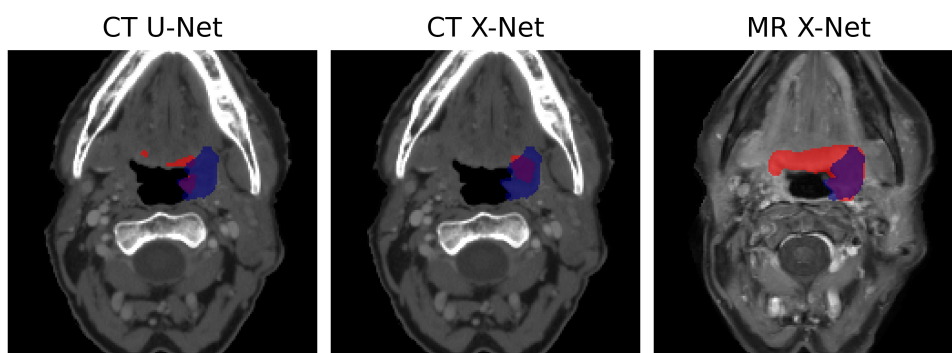


Figure A.1: Ground truth (blue) and predicted GTV (red) masks for patient ANON242. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.08, 0.24 (CT) and 0.55 (MR).

ANON590

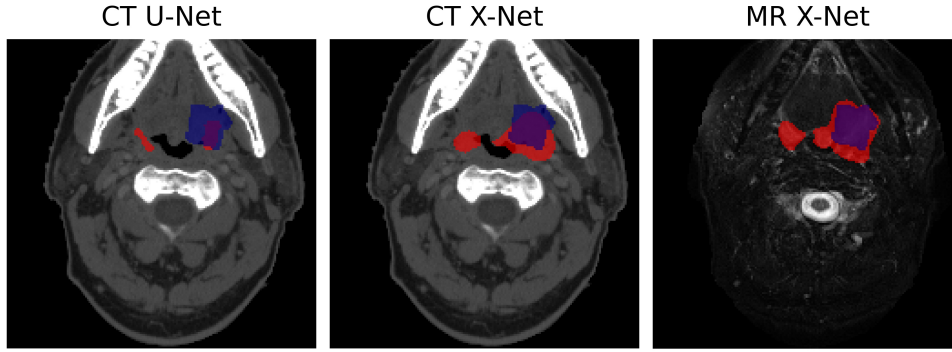


Figure A.2: Ground truth (blue) and predicted GTV (red) masks for patient ANON590. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.28, 0.48 (CT) and 0.60 (MR).

ANON690

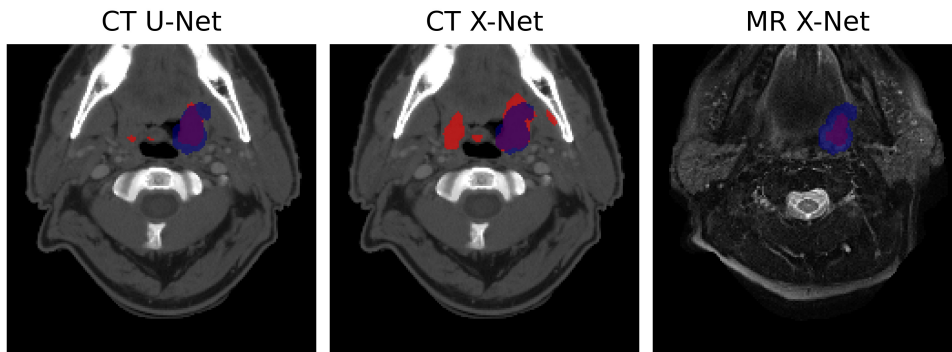


Figure A.3: Ground truth (blue) and predicted GTV (red) masks for patient ANON690. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.35, 0.45 (CT) and 0.52 (MR).

ANON866

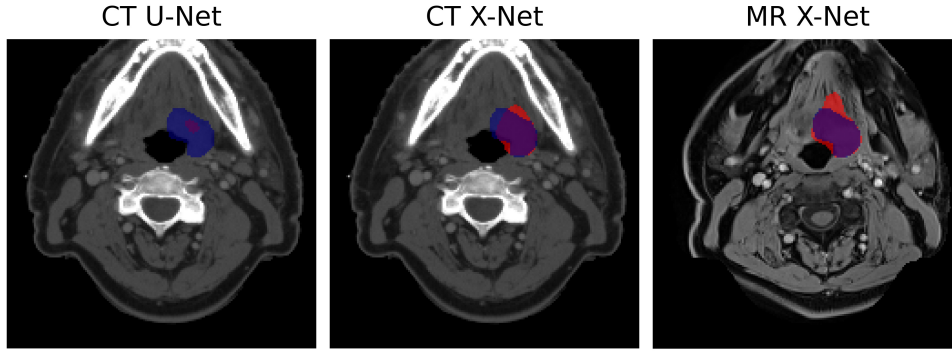


Figure A.4: Ground truth (blue) and predicted GTV (red) masks for patient ANON866. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.15, 0.52 (CT) and 0.46 (MR).

ANON1132

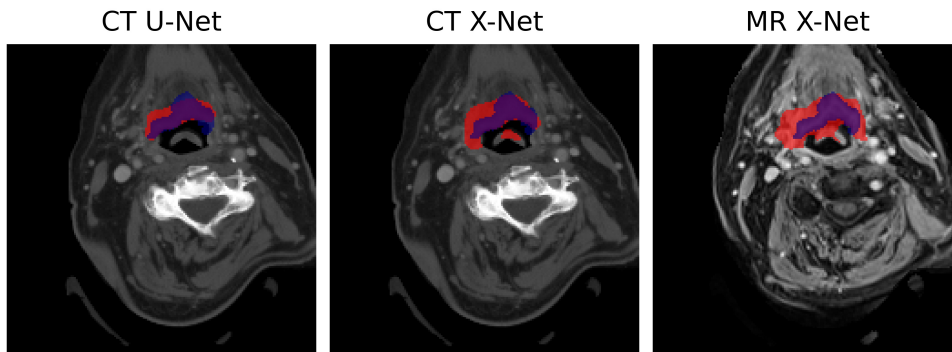


Figure A.5: Ground truth (blue) and predicted GTV (red) masks for patient ANON1132. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.53, 0.56 (CT) and 0.54 (MR).

ANON1677

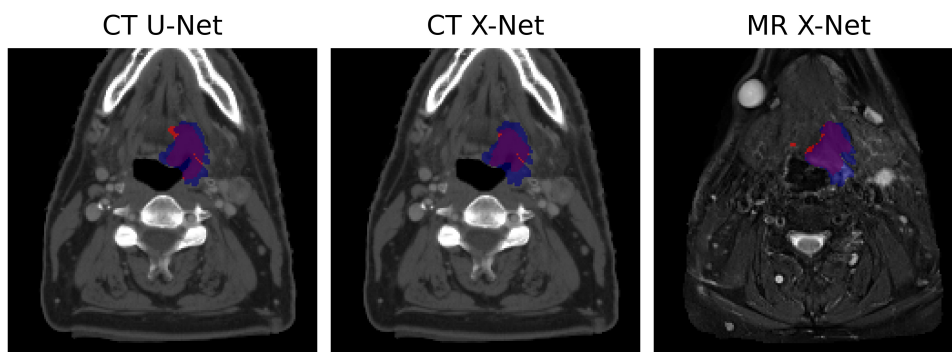


Figure A.6: Ground truth (blue) and predicted GTV (red) masks for patient ANON1677. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.61, 0.58 (CT) and 0.65 (MR).

ANON1783

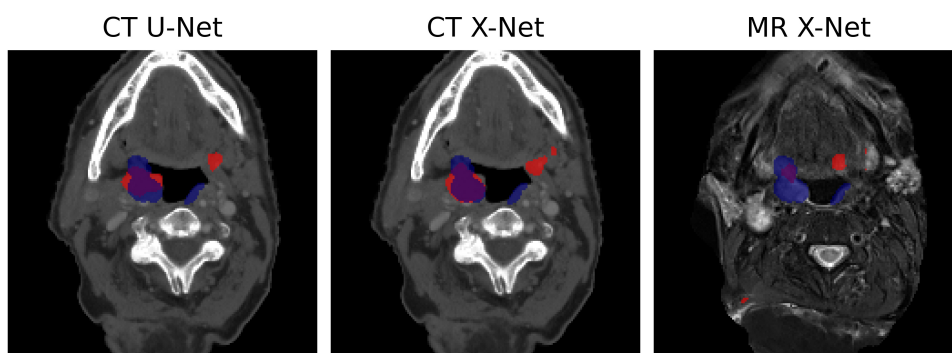


Figure A.7: Ground truth (blue) and predicted GTV (red) masks for patient ANON1783. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.32, 0.44 (CT) and 0.11 (MR).

ANON2453

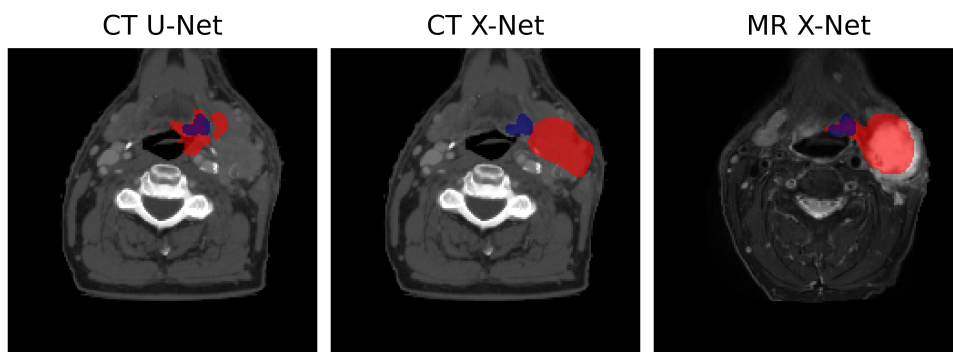


Figure A.8: Ground truth (blue) and predicted GTV (red) masks for patient ANON2453. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.20, 0.02 (CT) and 0.14 (MR).

ANON2834

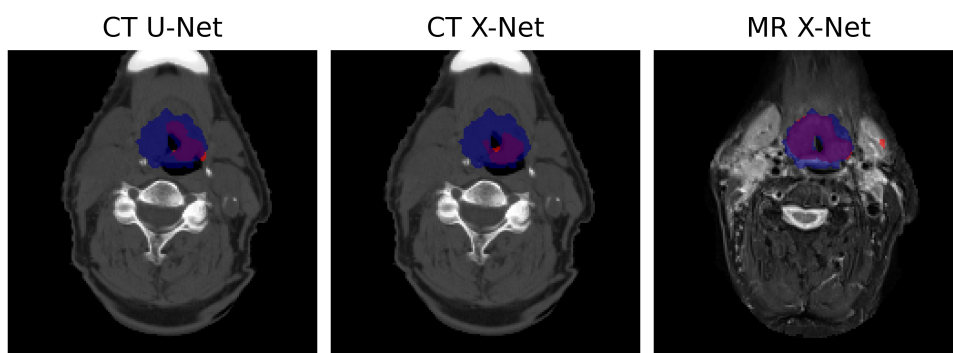


Figure A.9: Ground truth (blue) and predicted GTV (red) masks for patient ANON2834. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.29, 0.31 (CT) and 0.75 (MR).

ANON2865

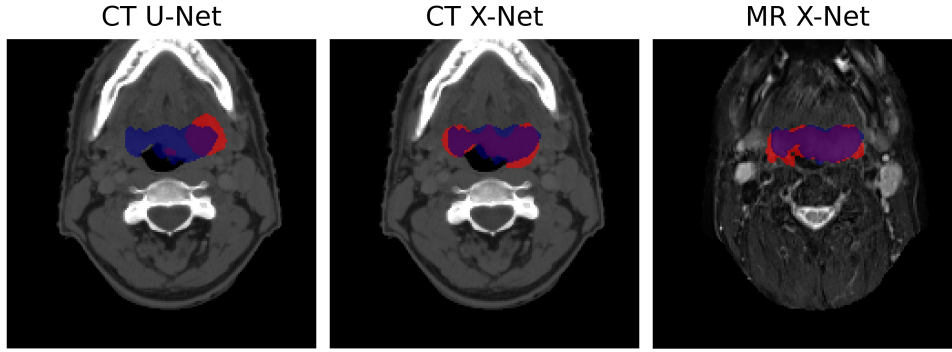


Figure A.10: Ground truth (blue) and predicted GTV (red) masks for patient ANON2865. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.49, 0.68 (CT) and 0.81 (MR).

ANON2971

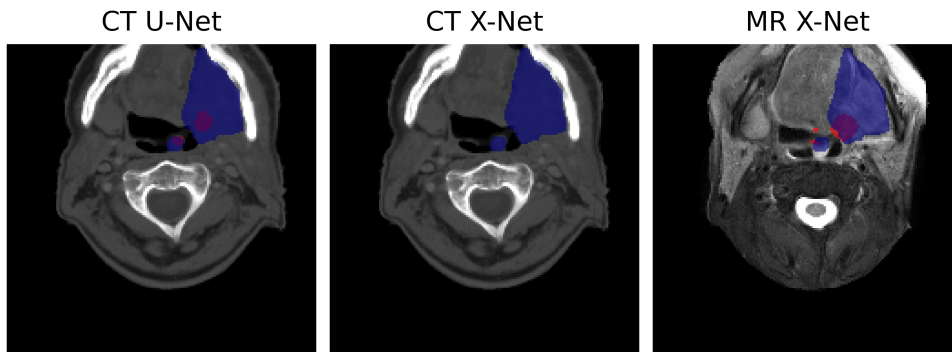


Figure A.11: Ground truth (blue) and predicted GTV (red) masks for patient ANON2971. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.06, 0.0 (CT) and 0.2 (MR).

ANON3146

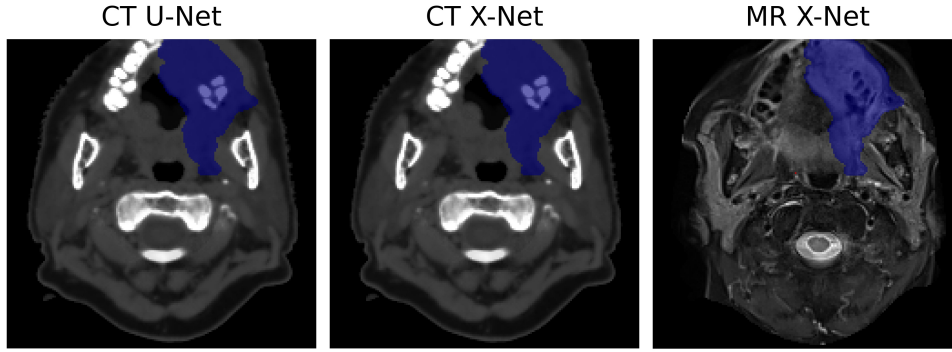


Figure A.12: Ground truth (blue) and predicted GTV (red) masks for patient ANON3146. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.0, 0.0 (CT) and 0.0 (MR).

ANON3207

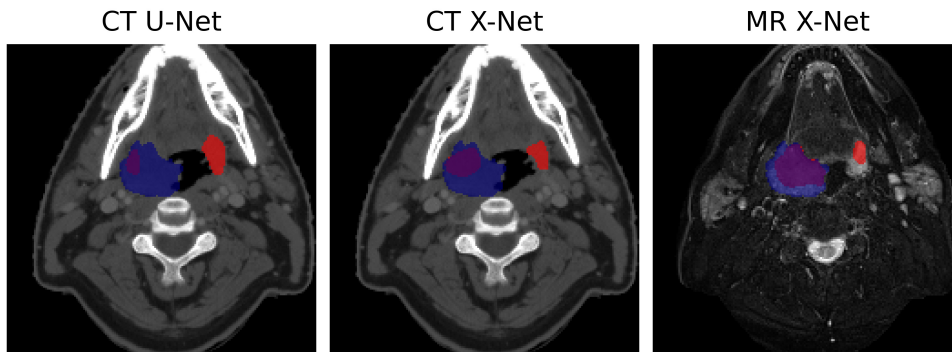


Figure A.13: Ground truth (blue) and predicted GTV (red) masks for patient ANON3207. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.26, 0.34 (CT) and 0.69 (MR).

ANON3435

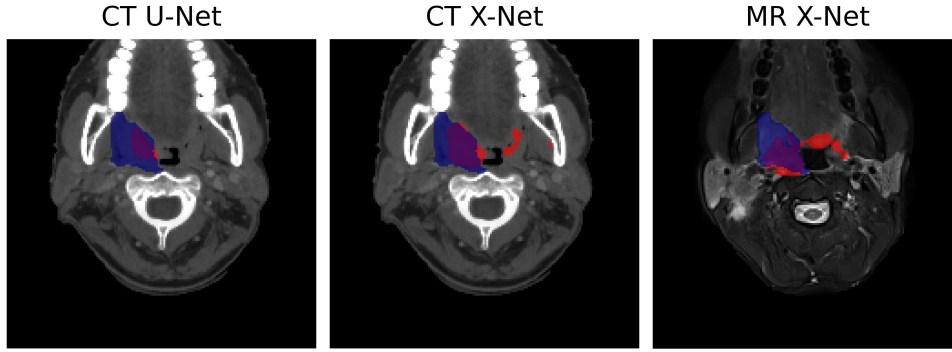


Figure A.14: Ground truth (blue) and predicted GTV (red) masks for patient ANON3435. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.42, 0.61 (CT) and 0.58 (MR).

ANON3620

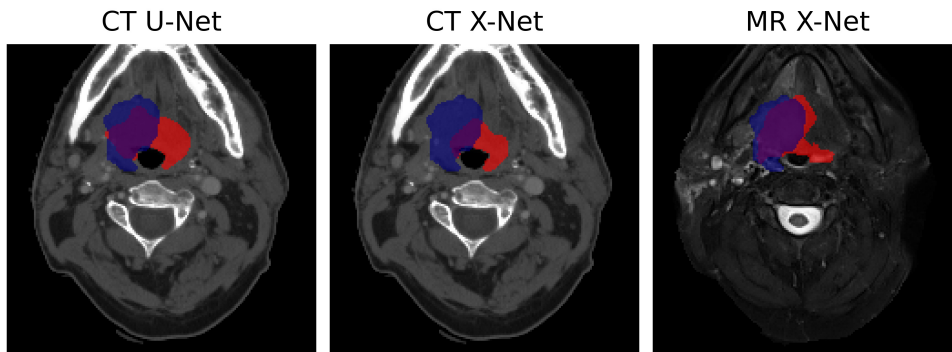


Figure A.15: Ground truth (blue) and predicted GTV (red) masks for patient ANON3620. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.3, 0.14 (CT) and 0.7 (MR).

ANON3647

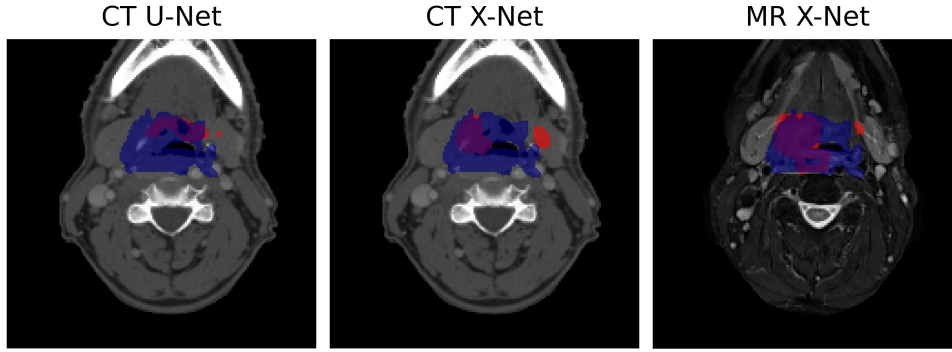


Figure A.16: Ground truth (blue) and predicted GTV (red) masks for patient ANON3647. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.12, 0.35 (CT) and 0.51 (MR).

ANON3648

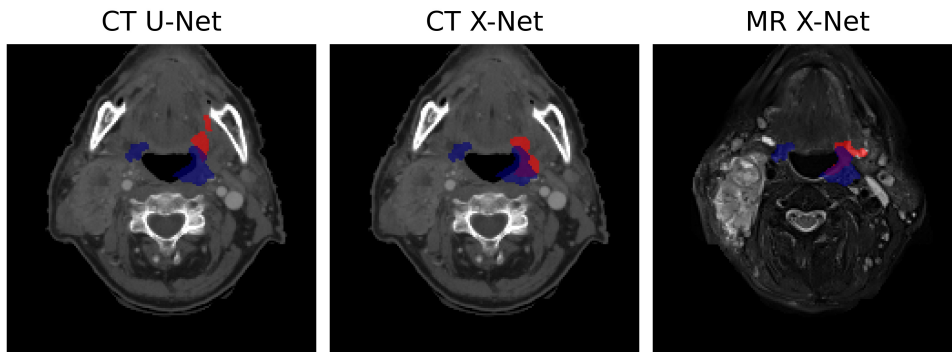


Figure A.17: Ground truth (blue) and predicted GTV (red) masks for patient ANON3648. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.25, 0.31 (CT) and 0.44 (MR).

ANON4232

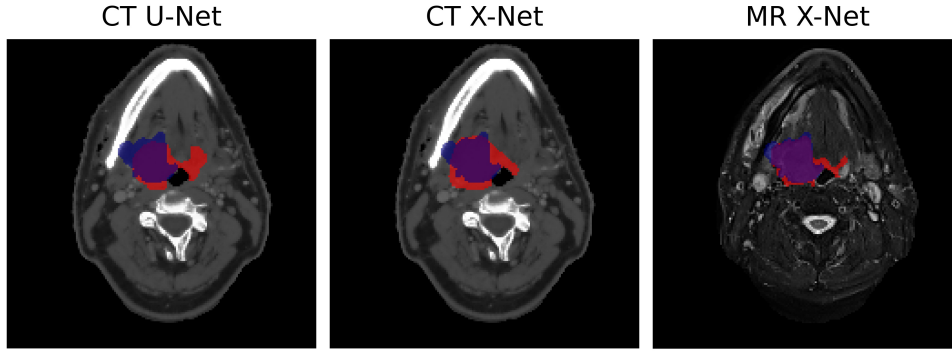


Figure A.18: Ground truth (blue) and predicted GTV (red) masks for patient ANON4232. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.63, 0.66 (CT) and 0.75 (MR).

ANON4297

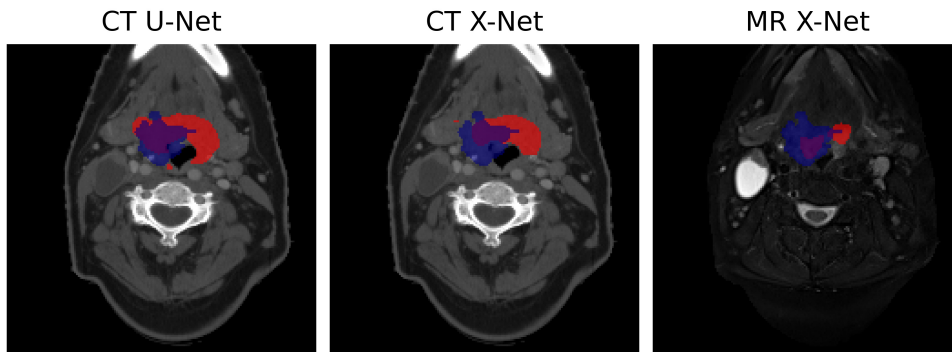


Figure A.19: Ground truth (blue) and predicted GTV (red) masks for patient ANON4297. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.46, 0.43 (CT) and 0.26 (MR).

ANON4641

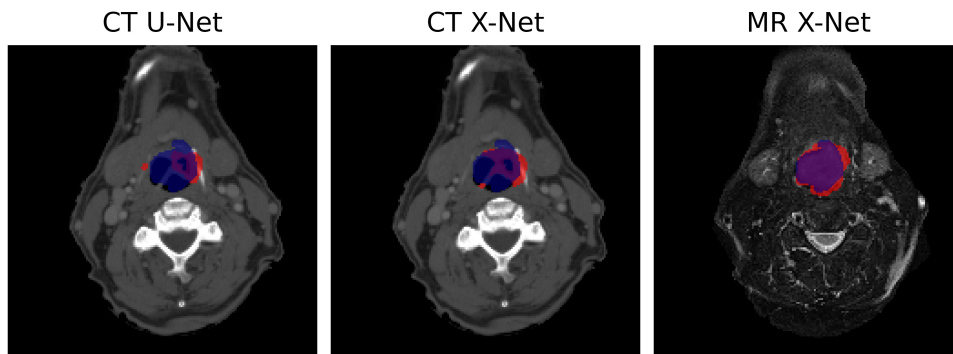


Figure A.20: Ground truth (blue) and predicted GTV (red) masks for patient ANON4641. Dice scores for the predictions of the U-Net and the X-Net (CT and MR) are respectively 0.28, 0.37 (CT) and 0.68 (MR).

References

- [1] Laurent Jacques and Christophe De Vleeschouwer. *LELEC2885 Image processing and computer vision*. Université catholique de Louvain, 2021.
- [2] Tom M Mitchell and Machine Learning. “McGraw-Hill”. In: *New York* (1997), pp. 154–200.
- [3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [4] Stuart Russel and Peter Norvig. *Artificial Intelligence: A modern approach*. Third Ed. Pearson Education, Inc, 2013.
- [5] Gordon E. Moore. “Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff.” In: *IEEE Solid-State Circuits Society Newsletter* 11.3 (2006), pp. 33–35. DOI: 10.1109/N-SSC.2006.4785860.
- [6] Richard E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 2015. ISBN: 9781400874668. DOI: doi:10.1515/9781400874668. URL: <https://doi.org/10.1515/9781400874668>.
- [7] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.
- [8] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow*. O’Reilly Media, Inc., 2019. ISBN: 9781492032649.
- [9] Ana Barragán-Montero et al. “Artificial intelligence and machine learning for medical imaging: A technology review”. In: *Physica Medica* 83 (2021), pp. 242–256.
- [10] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.
- [11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [12] Yuanyuan Liu et al. “Flower classification via convolutional neural network”. In: *2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA)*. 2016, pp. 110–116. DOI: 10.1109/FSPMA.2016.7818296.
- [13] Ossama Abdel-Hamid et al. “Convolutional neural networks for speech recognition”. In: *IEEE/ACM Transactions on audio, speech, and language processing* 22.10 (2014), pp. 1533–1545.
- [14] Alexis Conneau et al. “Very deep convolutional networks for natural language processing”. In: *arXiv preprint arXiv:1606.01781* 2.1 (2016).
- [15] Dana Lahat, Tülay Adalı, and Christian Jutten. “Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects”. In: *Proceedings of the IEEE* 103.9 (2015), pp. 1449–1477. DOI: 10.1109/JPROC.2015.2460697.
- [16] Kedar Potdar, Taher S. Pardawala, and Chinmay D. Pai. “A comparative study of categorical variable encoding techniques for neural network classifiers”. In: *International Journal of Computer Applications* 175.4 (2017), pp. 7–9. DOI: 10.5120/ijca2017915495.
- [17] R.H. Ranger. “Transmission and Reception of Photoradiograms”. In: *Proceedings of the Institute of Radio Engineers* 14.2 (1926), pp. 161–180. DOI: 10.1109/JRPROC.1926.221019.
- [18] W. S. Boyle and G. E. Smith. “Charge coupled semiconductor devices”. In: *The Bell System Technical Journal* 49.4 (1970), pp. 587–593. DOI: 10.1002/j.1538-7305.1970.tb01790.x.
- [19] Eike Rietzel, Tinsu Pan, and George T. Y. Chen. “Four-dimensional computed tomography: Image formation and clinical protocol”. In: *Medical Physics* 32.4 (2005), pp. 874–889. DOI: <https://doi.org/10.1118/1.1869852>.

- [20] KR1442 Chowdhary. “Natural language processing”. In: *Fundamentals of artificial intelligence* (2020), pp. 603–649.
- [21] Luc Vandendorpe. *LELEC1360 Télécommunication*. Université catholique de Louvain, 2019.
- [22] Noriki Nishida and Hideki Nakayama. “Multimodal gesture recognition using multi-stream recurrent neural network”. In: *Image and Video Technology*. Springer. 2015, pp. 682–694.
- [23] Dhanesh Ramachandram and Graham W Taylor. “Deep multimodal learning: A survey on recent advances and trends”. In: *IEEE signal processing magazine* 34.6 (2017), pp. 96–108.
- [24] Hugh F Durrant-Whyte. “Sensor models and multisensor integration”. In: *Autonomous robot vehicles*. Springer, 1990, pp. 73–89.
- [25] Federico Castanedo. “A review of data fusion techniques”. In: *The scientific world journal* 2013 (2013).
- [26] Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. “Multimodal deep learning for biomedical data fusion: a review”. In: *Briefings in Bioinformatics* 23.2 (Jan. 2022). ISSN: 1477-4054. DOI: 10.1093/bib/bbab569.
- [27] Pradeep K Atrey et al. “Multimodal fusion for multimedia analysis: a survey”. In: *Multimedia systems* 16.6 (2010), pp. 345–379.
- [28] WH Adams et al. “Semantic indexing of multimedia content using visual, audio, and text cues”. In: *EURASIP Journal on Advances in Signal Processing* 2003.2 (2003), pp. 1–16.
- [29] Yi Wu et al. “Optimal multimodal fusion for multimedia data analysis”. In: *Proceedings of the 12th annual ACM international conference on Multimedia*. 2004, pp. 572–579.
- [30] Joshua B Tenenbaum, Vin de Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *science* 290.5500 (2000), pp. 2319–2323.
- [31] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Vol. 1. Springer, 2007.
- [32] Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.
- [33] Huma Lodhi et al. “Text classification using string kernels”. In: *Journal of machine learning research* 2.Feb (2002), pp. 419–444.
- [34] Mehmet Gönen and Ethem Alpaydm. “Multiple kernel learning algorithms”. In: *The Journal of Machine Learning Research* 12 (2011), pp. 2211–2268.
- [35] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis Philippe Morency. “Multimodal Machine Learning: A Survey and Taxonomy”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2 Feb. 2019), pp. 423–443. ISSN: 19393539. DOI: 10.1109/TPAMI.2018.2798607.
- [36] Charles Sutton, Andrew McCallum, et al. “An introduction to conditional random fields”. In: *Foundations and Trends® in Machine Learning* 4.4 (2012), pp. 267–373.
- [37] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation Learning: A Review and New Perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50.
- [38] *Sparse matrix*. Mar. 2022. URL: https://en.wikipedia.org/wiki/Sparse_matrix.
- [39] Bruno A Olshausen and David J Field. “Emergence of simple-cell receptive field properties by learning a sparse code for natural images”. In: *Nature* 381.6583 (1996), pp. 607–609.
- [40] Nitish Srivastava and Russ R Salakhutdinov. “Multimodal learning with deep boltzmann machines”. In: *Advances in neural information processing systems* 25 (2012).
- [41] Yelin Kim, Honglak Lee, and Emily Mower Provost. “Deep learning for robust feature generation in audiovisual emotion recognition”. In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 3687–3691.
- [42] Andrea Frome et al. “Devise: A deep visual-semantic embedding model”. In: *Advances in neural information processing systems* 26 (2013).

- [43] Jason Weston, Samy Bengio, and Nicolas Usunier. “Wsabie: Scaling up to large vocabulary image annotation”. In: *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.
- [44] Ankush Gupta, Yashaswi Verma, and C Jawahar. “Choosing linguistics over vision to describe images”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 26. 1. 2012, pp. 606–612.
- [45] Andrei Barbu et al. “Video in sentences out”. In: *arXiv preprint arXiv:1204.2742* (2012).
- [46] Takaki Makino et al. “Recurrent neural network transducer for audio-visual speech recognition”. In: *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE. 2019, pp. 905–912.
- [47] Yukun Zhu et al. “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 19–27.
- [48] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).
- [49] Avrim Blum and Tom Mitchell. “Combining labeled and unlabeled data with co-training”. In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 92–100.
- [50] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. “A survey of transfer learning”. In: *Journal of Big data* 3.1 (2016), pp. 1–40.
- [51] Ekaterina Shutova, Douwe Kiela, and Jean Maillard. “Black holes and white rabbits: Metaphor identification with visual features”. In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. 2016, pp. 160–170.
- [52] Janarthanan Rajendran et al. “Bridge correlational neural networks for multilingual multimodal representation learning”. In: *arXiv preprint arXiv:1510.03519* (2015).
- [53] Shih-Cheng Huang et al. “Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines”. In: *NPJ digital medicine* 3.1 (2020), pp. 1–9.
- [54] Kuan Liu et al. “Learn to combine modalities in multimodal deep learning”. In: *arXiv preprint arXiv:1805.11730* (2018).
- [55] Douwe Kiela et al. “Efficient large-scale multi-modal classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.
- [56] John Edison Arevalo Ovalle et al. “Gated Multimodal Units for Information Fusion.” In: *ICLR (Workshop)*. 2017.
- [57] Akira Fukui et al. “Multimodal compact bilinear pooling for visual question answering and visual grounding”. In: *arXiv preprint arXiv:1606.01847* (2016).
- [58] Jordan Yap, William Yolland, and Philipp Tschandl. “Multimodal skin lesion classification using deep learning”. In: *Experimental dermatology* 27.11 (2018), pp. 1261–1267.
- [59] Kaiming He et al. “Identity mappings in deep residual networks”. In: *European conference on computer vision*. Springer. 2016, pp. 630–645.
- [60] Juan Song et al. “An effective multimodal image fusion method using MRI and PET for Alzheimer’s disease diagnosis”. In: *Frontiers in Digital Health* 3 (2021), p. 637386.
- [61] Kim-Han Thung, Pew-Thian Yap, and Dinggang Shen. “Multi-stage diagnosis of Alzheimer’s disease with incomplete multimodal data via multi-task deep learning”. In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 160–168.
- [62] Natalia Neverova et al. “Multi-scale deep learning for gesture detection and localization”. In: *European conference on computer vision*. Springer. 2014, pp. 474–490.
- [63] Janani Venugopalan et al. “Multimodal deep learning models for early detection of Alzheimer’s disease stage”. In: *Scientific reports* 11.1 (2021), pp. 1–13.

- [64] Seyed Aghil Hooshmand et al. “A multimodal deep learning-based drug repurposing approach for treatment of COVID-19”. In: *Molecular diversity* 25.3 (2021), pp. 1717–1730.
- [65] Dongdong Zhang et al. “Combining structured and unstructured data for predictive models: a deep learning approach”. In: *BMC medical informatics and decision making* 20.1 (2020), pp. 1–11.
- [66] Yagya Raj Pandeya and Joonwhoan Lee. “Deep learning-based late fusion of multimodal information for emotion classification of music video”. In: *Multimedia Tools and Applications* 80.2 (2021), pp. 2887–2905.
- [67] Emilie Morvant, Amaury Habrard, and Stéphane Ayache. “Majority vote of diverse classifiers for late fusion”. In: *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*. Springer. 2014, pp. 153–162.
- [68] Jean-Francis Roy, François Laviolette, and Mario Marchand. “From PAC-Bayes bounds to quadratic programs for majority votes”. In: *ICML*. 2011.
- [69] Islam Reda et al. “Deep learning role in early diagnosis of prostate cancer”. In: *Technology in cancer research & treatment* 17 (2018), p. 1533034618775530.
- [70] Eléonore Liefbrig, Edmond Sterpin, and John Aldo Lee. “Automatic Delineation of Primary Tumor Volumes from MR Images in Head and Neck Cancer Patients with Deep Learning”. In: ().
- [71] Haochen Mei et al. “Automatic segmentation of gross target volume of nasopharynx cancer using ensemble of multiscale deep neural networks with spatial attention”. In: *Neurocomputing* 438 (2021), pp. 211–222.
- [72] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.
- [73] Dakai Jin et al. “Accurate esophageal gross tumor volume segmentation in PET/CT using two-stream chained 3D deep network fusion”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 182–191.
- [74] Jinzhong Yang et al. “A multimodality segmentation framework for automatic target delineation in head and neck radiotherapy”. In: *Medical physics* 42.9 (2015), pp. 5310–5320.
- [75] Fabian Isensee et al. “nnu-net: Self-adapting framework for u-net-based medical image segmentation”. In: *arXiv preprint arXiv:1809.10486* (2018).
- [76] Fabian Isensee et al. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18.2 (2021), pp. 203–211.
- [77] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 565–571.
- [78] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [79] *Deming regression*. Aug. 2022. URL: https://en.wikipedia.org/wiki/Deming_regression.
- [80] Fan Tang et al. “Postoperative glioma segmentation in CT image using deep feature fusion model guided by multi-sequence MRIs”. In: *European Radiology* 30.2 (2020), pp. 823–832.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl