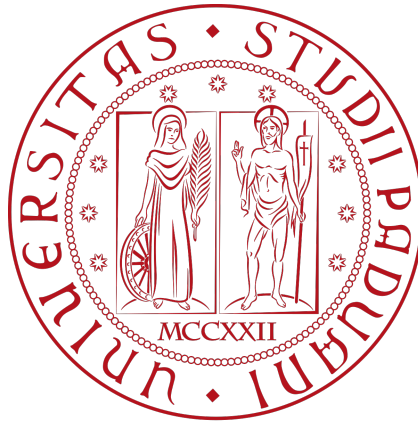


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI BIOLOGIA

Corso di Laurea magistrale in

Quantitative and Computational Biosciences



TESI DI LAUREA

**Computer Vision algorithms
for microscopy images**

Relatore: *Prof. Gabriele Sales*

Dipartimento di Biologia

Correlatore: *Prof.ssa Nicoletta Plotegher*

Dipartimento di Biologia

Laureando: *Matteo Veronese*

ANNO ACCADEMICO 2025/2026

1 Abstract

Deep learning in computer vision has achieved remarkable results in visual recognition; however, its application to biological microscopy remains far more challenging than to natural images. This is due to low signal-to-noise ratios, uneven illumination, and dense, heterogeneous cellular environments that contain multiple similar organelles. Models capable of reliably interpreting these structures play a crucial role in diagnostics, as well as in genomic and molecular studies, where precise morphological characterization can support the identification and study of pathological conditions.

In this thesis, We investigate two complementary computer vision approaches applied to Transmission Electron Microscopy (TEM) images. First, we show that a standard, supervised U-Net architecture can achieve robust performance in mitochondrial semantic segmentation, highlighting that its main limitation lies not in the model itself, but in the scarcity of well-annotated data. We then explore a fully self-supervised alternative, training a ResNet-50 backbone with a self-supervised approach on images from patients affected by Gaucher disease (GD) —a lysosomal storage disorder characterized by high phenotypic heterogeneity— to learn general-purpose representations directly from unlabeled data. We show that the resulting trained backbone is able to capture meaningful morphological features and clinically relevant patterns without explicit supervision, demonstrating the ability not only to distinguish between different subcellular structures but also to capture the underlying signals that allow patient groups to be differentiated according to disease severity.

Contents

1	Abstract	ii
2	Introduction	1
2.1	Supervised Learning	2
2.1.1	From Fully Convolutional Networks to U-Net	3
2.1.2	U-Net: Encoder-Decoder and Skip Connections	5
2.2	Self-Supervised Learning	6
2.2.1	Biological Background	6
2.2.2	Self-Supervised Learning in Computer Vision	9
2.2.3	SimSiam: a Non-Contrastive Joint-Embedding Architecture	11
3	Material and Methods	14
3.1	Supervised Learning: Mitochondria Segmentation	14
3.1.1	Dataset and Data Pre-Processing	14
3.1.2	Data Augmentation	16
3.1.3	Model Architecture and Training Configuration	17
3.2	Self-Supervised	19
3.2.1	Dataset and Data Pre-Processing	19
3.2.2	Data Augmentation	21
3.2.3	Training Configuration	24
3.2.4	Downstream Analysis	26
3.3	Experimental setup	31
4	Results and Discussion	32
4.1	Supervised Learning: Mitochondria Segmentation	32
4.2	Self-Supervised Learning	33
4.2.1	General pre-training and fine-tuning on the Gaucher dataset	33
4.2.2	Pre-training on Gaucher dataset and qualitative analysis of the latent space	34

4.2.3	Assessment of generalization using LOPO cross-validation	37
5	Conclusions	45
	Bibliography	48

2 Introduction

Computer Vision is a branch of Artificial Intelligence (AI) that focuses on developing algorithms capable of processing, extracting and interpreting high-level information from digital images or video streams. In the healthcare sector, this involves applying AI techniques to medical images and videos—such as X-rays, computed tomography (CT) scans, magnetic resonance imaging (MRI), endoscopic footage or TEM images—in order to extract clinically relevant information and support a wide range of diagnostic and analytical tasks.

Within this broad landscape, this thesis focuses on the use of images acquired via Transmission Electron Microscopy (TEM), an advanced technique that uses an electron beam to examine biological samples, enabling detailed observation of cellular structures at nanometer resolution.

Unlike natural images, cellular TEM images present unique challenges for AI models. Indeed, cellular organelles exhibit extreme variability in appearance depending on the plane of tissue sectioning, resulting in considerable morphological heterogeneity: organelles of the same type—whether located within the same cell or across different cells—can vary significantly in shape and size. Added to this is the fact that these structures are not always clearly visible or easily distinguishable from one another; in some cases, they may be in close proximity or even interconnected, making their identification even more complex.

A further source of complexity stems from the nature of TEM images, where each pixel encodes a single scalar intensity value rather than color information. Different cellular structures often exhibit very similar intensity levels, making it difficult to distinguish them on the basis of intensity alone. Added to this is the almost constant presence of background noise, due to the high sensitivity of biological samples to the electron beam used for imaging.

This marked variability would ideally require models trained on large and representative datasets. However, this domain faces a structural limitation: the preparation, acquisition, and potential annotation of TEM images require a high level of specialist expertise and entail significant costs, resulting in a

scarcity of data available for training.

Despite these challenges, the various available TEM datasets are utilized in this work through two distinct methodological approaches to artificial intelligence.

On the one hand, the task of semantic segmentation of mitochondria is tackled in a supervised manner. Semantic segmentation consists in classifying each pixel of an image according to the object or structure it belongs to, thus producing a pixel-level map rather than a single label for the whole image. So in this context, the model uses carefully annotated images to learn the pixel-by-pixel distinction between mitochondrial structures and the background, thereby enabling an assessment of its performance and generalization ability in contexts characterized by limited data availability. On the other hand, a self-supervised approach based on the SimSiam algorithm is adopted, which eliminates the need for manual annotations. The aim of this second experiment is twofold: first, to understand whether the model can capture the information structure of TEM images despite the noise and visual variability described above; second, to verify whether the morphological representations thus learned are sufficiently rich to allow patients to be categorized into distinct clinical groups, classifying them according to their pathological phenotype and the severity of their disease.

For this reason, in the following chapters, the various topics will be dealt with separately for each of the two types of task.

2.1 Supervised Learning

Supervised-Learning is a training approach in which the model requires pairs of input-output data to adjust its parameters. Its aim is to learn the optimal function capable of accurately mapping inputs to their corresponding outputs, while minimizing the prediction error.

In this section, the proposed method is applied to the semantic segmentation of mitochondria [1], a task in which the objective is to assign each pixel in the image to one of two mutually exclusive classes: mitochondrion or background.

Because this is a semantic segmentation task, the model classifies the category to which each individual pixel belongs, without distinguishing between separate mitochondrial structures.

Semantic mitochondrial segmentation is one of the most common tasks in TEM image analysis, as a wide variety of quantitative information can be extracted from mitochondria. This includes morphometric quantities such as the area of the inner matrix and the intermembrane space, mitochondrial perimeter, the number of mitochondria per cell, the number of mitochondrial cristae, and the number of connections between adjacent organelles. This information is particularly relevant for both basic research and clinical settings: mitochondrial morphology and count are strongly correlated with severe pathologies such as cancer [2], Parkinson’s disease [3], and other neurodegenerative disorders [4]. Consequently, automated and accurate mitochondrial segmentation acts as a key enabler for a wide range of downstream analyses, which would otherwise be significantly slower and more resource-intensive if performed manually.

The main limitation of the supervised approach, however, lies in its very nature: the effectiveness of deep learning models in this task depends heavily on the availability of large volumes of accurately labeled data. Manual annotation of images at the single-pixel level is a highly time-consuming and labor-intensive process, thus representing the primary bottleneck for model training.

2.1.1 From Fully Convolutional Networks to U-Net

Before the advent of Fully Convolutional Networks (FCNs), deep neural network-based approaches to semantic segmentation typically relied on convolutional classifiers applied locally using a sliding-window approach: for each pixel in the image, a small surrounding patch was extracted and classified individually by a CNN to determine its class membership. Although effective, this approach was extremely computationally expensive, as adjacent patches overlapped extensively and the network was forced to recalculate the same convolutional features multiple times for nearly identical image regions, making the process slow and poorly scalable on large images. Fully Convolutional Networks [5]

overcame this limitation by reformulating the entire problem in an end-to-end manner. By replacing the fully-connected layers of classical classification networks (AlexNet, VGG16, GoogLeNet) entirely with convolutional layers, the network directly produces a prediction map for the entire image in a single pass, sharing the feature calculation between neighboring pixels rather than repeating it for each individual patch. This architectural choice, besides drastically improving computational efficiency, allows the network to process images of arbitrary size and to produce, via upsampling operations, a segmentation map of the same size as the input image.

Building on this seminal work, several improvements have been proposed, motivated by specific limitations of FCNs. ParseNet [6] observed that FCNs tend to lose the global context of the image in deeper layers, proposing a module that integrates a global context vector —obtained via pooling and normalization— with the original feature maps. A different approach was proposed by Noh, Hong, and Han [7], who introduced a network comprising a convolutional encoder and a symmetric deconvolutional decoder, capable of reconstructing pixel-by-pixel probability maps from region proposals generated by an object detection model. It is within this context that U-Net [8] emerged, proposed specifically for the segmentation of biomedical images, particularly microscopy images. Unlike the original FCNs, which reconstruct spatial resolution via relatively simple upsampling, U-Net introduces a decoder that is fully symmetric to the encoder and equally deep. It is connected to the encoder via skip connections that concatenate —rather than summing, as was the case in FCNs— the encoder’s high-resolution feature maps with the corresponding ones from the decoder. This mechanism allows fine spatial information to be preserved across multiple scales, integrating, at each stage of reconstruction, both the abstract semantic content learned in the deeper layers and the geometric details of the original structures —an aspect that is particularly critical when the aim is to precisely delineate the contours of small structures, as in the case of the mitochondria discussed in this thesis.

A further factor that contributes to U-Net’s enduring popularity, particularly in the biomedical field, is its effectiveness even when annotated data

is scarce. The original authors themselves validated the architecture by successfully training it on datasets comprising a small number of images, utilizing a data augmentation pipeline to artificially increase the variability of the training set. This characteristic makes U-Net particularly well-suited to contexts, such as the one addressed in this thesis, where the availability of annotated images is extremely limited, and explains why, years after its introduction, it remains one of the most robust and widely used benchmark models for biomedical image segmentation.

2.1.2 U-Net: Encoder-Decoder and Skip Connections

In this thesis, therefore, the U-Net architecture was adopted, characterized by the presence of a contracting path and an expansive path. The first part, known as the *encoder*, has the typical structure of a convolutional neural network (CNN): it consists of a sequence of convolution and max-pooling layers, interspersed with non-linear activation functions. During this phase, the model progressively compresses the spatial information of the input image, projecting it into a low-dimensional representation space. The second part, the *decoder*, performs the opposite function: starting from this compressed representation, it reconstructs the prediction map at the single-pixel level. The output generated by the decoder is a binary mask in which white pixels indicate the presence of mitochondria, while black pixels represent the background.

To train the model on the mitochondrial segmentation task, each TEM image is paired with a corresponding binary mask, where pixels belonging to mitochondria are assigned a value of 1, and background pixels are set to 0. By repeatedly examining these input-output pairs during training, the model progressively learns the distinctive features of mitochondria, learning to differentiate them from the rest of the background (cytoplasm, endoplasmic reticulum, etc.). Further details on the architecture, the hyperparameters used during training and the dataset employed will be provided in the Material and Methods section.

2.2 Self-Supervised Learning

The previous section described the classic mechanism of supervised learning, in which the dataset consists of input-output pairs and the model must find the function $F(x)$ that links the two. In self-supervised learning, by contrast, the dataset has no labels but consists solely of a series of unlabeled samples. In the medical field, this aspect is particularly advantageous: as mentioned earlier, the main obstacle to training supervised models is the need for a sufficiently large and accurately annotated dataset, but the manual annotation process is typically costly and time-consuming. Self-supervised learning, on the other hand, seeks to extract the information contained in the images without using explicit labels: the supervision signal is intrinsic to the dataset itself, which significantly reduces the costs and time required for its preparation. However, this advantage comes at a cost: as the learning task is more complex and less constrained than in the supervised case, these models typically require a larger number of unlabeled images to achieve comparable performance [9]. This requirement is further accentuated in the presence of images with high intrinsic variability, as in the case of the TEM images discussed earlier, where morphological heterogeneity and background noise make the representation task even more challenging.

2.2.1 Biological Background

This type of training is applied to TEM images relating to GD, a condition caused by mutations in the *GBA1* gene, which encodes the lysosomal enzyme glucocerebrosidase [10]. This enzyme is responsible for the breakdown of glucosylceramide, a membrane lipid that serves as the biosynthetic precursor of numerous complex glycosphingolipids, which are essential for the structure of cell membranes and for cellular signaling processes. In the presence of mutations in *GBA1*, enzymatic activity is reduced or absent, preventing the normal hydrolysis of glucosylceramide. As a result, this substrate progressively accumulates within the lysosomes of various tissues and organs throughout the

body, primarily in the liver, spleen, and bone marrow [10]. This accumulation impairs overall lysosomal function, particularly affecting the cell's ability to correctly degrade proteins and other cellular substrates, resulting in the dysfunction of the main intracellular degradation pathways [11].

Mutations in the *GBA1* gene are of clinical interest beyond GD alone: indeed, heterozygosity for these mutations is the main known genetic risk factor for Parkinson's disease (PD) [12], to the extent that between 3% and 25% of patients with Parkinson's disease are carriers of a *GBA1* variant [13]. However, only a small percentage of individuals carrying *GBA1* variants actually develop parkinsonism during their lifetime, indicating an overall low penetrance. This link makes the study of cellular alterations induced by *GBA1* mutations relevant not only to understanding GD itself, but also as a potential window into pathogenic mechanisms shared with other neurodegenerative disorders.

Although GD is a monogenic disorder, it manifests clinically with a wide range of phenotypic variability. From a clinical perspective, it is classically divided into three main forms [13]: non-neuronopathic GD (*GD1*), acute neuronopathic GD (*GD2*), and chronic neuronopathic GD (*GD3*). Historically, the *GD1* form—the most frequent of the three—was defined by the strict absence of neurological involvement; clinical manifestations typically included hematological abnormalities, such as anemia and thrombocytopenia (a reduction in platelet count below homeostatic levels), as well as skeletal problems. However, recent literature indicates that this rigid distinction is evolving, as a spectrum of neurological signs and symptoms (e.g. Parkinsonism and peripheral neuropathy) has been increasingly recognized in adult patients classified with GD1 [14]. Even within this single form, clinical variability is marked: severely symptomatic patients are observed alongside completely asymptomatic individuals, who are identified only occasionally through screening.

The *GD2* subtype, on the other hand, is the most severe form. It is characterized by severe and rapidly progressive neurological involvement, which typically leads to the patient's death in early childhood.

Although the *GD3* form also presents with neurological involvement, it differs markedly from GD2 in having a significantly slower and more chronic

course. For this reason, from a clinical perspective, it is in some respects more comparable to *GD1*—with which it shares a more gradual progression of the disease— than to *GD2*. As will be detailed in the *Material and Methods* section, this greater clinical affinity between *GD1* and *GD3* provides the rationale for grouping the two subtypes into a single macro-class (*G1_G3*), distinct from the *G2* class, in order to categorize the input images into clinically relevant groups. Furthermore, the clinical presentation across all GD forms is strongly influenced by genetic modifiers whose prevalence varies depending on the population under consideration, contributing to further intrinsic variability within each specific phenotype.

However, a particularly significant aspect for the analysis conducted in this thesis is the high phenotypic heterogeneity observed even among individuals carrying the same mutation. Indeed, the correlation between genotype and phenotype is weak, most likely due to genetic or environmental modifiers that are not yet fully understood. This results in marked clinical and morphological variability even within the same severity category, such that two patients classified in the same clinical group may exhibit phenotypic characteristics that only partially overlap. The condition therefore does not manifest through clear-cut and unambiguous morphological boundaries between one severity level and another, but rather as a continuous spectrum of cellular alterations, in which certain characteristics found in a patient of a particular severity level may also be present, to varying degrees, in patients of different severity grades. This continuity, combined with intra-class variability, makes automated image analysis a particularly complex task: the model must be able to detect subtle and not always consistent morphological patterns within the same clinical group, rather than learning clear separation boundaries between classes.

Since the condition is driven by a monogenic mutation resulting in an enzyme deficiency, the gold standard for diagnosing GD relies on a *beta-glucosidase leukocyte* (BGL) blood test to measure enzyme activity. Additionally, the diagnosis can be definitively confirmed through genetic screening, by identifying the presence of the specific mutation in the *GBA1* gene through a blood sample [15]. While both procedures are relatively straightforward and minimally

invasive, being based on standard blood sampling, further diagnostic workup—such as a bone marrow biopsy, often performed to assess organ involvement or rule out other hematological conditions—can be considerably more invasive and costly.

Consequently, the objective of the AI model is to learn an effective latent representation capable of capturing only those morphological features relevant to describing the biological content of the image. The immediate goal is to classify each patient—based solely on their TEM images—into the clinical group to which they belong (healthy or affected, and in the latter case, stratified by severity) without resorting to any manual annotation or labeling. However, the scope of this work extends beyond merely assessing the model’s capacity to interpret complex visual data or to categorize patients. From a broader clinical perspective, this AI-driven approach is envisioned as serving a twofold purpose. On the one hand, it could act as a potential preliminary diagnostic screening tool: by automatically identifying disease-specific cellular alterations from TEM images, such a model could help flag high-risk patients, providing a morphological rationale for clinicians to proceed selectively with more specific diagnostic procedures—such as bone marrow analysis—reserving them for cases where the AI-based screening indicates a higher likelihood of disease. On the other hand, by learning to recognize the morphological patterns associated with each severity grade in a purely data-driven manner, the model may also contribute to a deeper understanding of the still largely unclear mechanisms underlying the disease, potentially highlighting subtle structural alterations that have not yet been characterized in the existing literature.

2.2.2 Self-Supervised Learning in Computer Vision

These self-supervised learning models can be described within the conceptual framework of *Energy-Based Models*, in which the objective is to assign high energy (high scalar values) to incompatible inputs and low energy (low scalar values) to compatible inputs [16]. Within this framework, the various approaches proposed in the literature can be classified into three broad families (Fig. 1), both generative and non-generative.

Generative Architectures In this type of model, the system learns to reconstruct a signal y from a signal x , conditioning the reconstruction via a latent variable z . In the context of images, z typically corresponds to a mask applied to x , which tells the model which portion of the original image is to be reconstructed. A representative example of this family is *Masked Autoencoders* [17], in which certain patches of the image are randomly masked and the missing pixels are subsequently reconstructed by the model. The reconstruction therefore takes place at the single-pixel level, and the loss is calculated directly in the input space.

Joint-Embedding Architectures In this case, two views are generated from the same image using data augmentation [18]. A shared encoder produces an embedding for each view, with the aim of obtaining similar representations for views of the same image. Unlike generative models, the loss is not calculated in the input space but in the embedding space: therefore, rather than a pixel-by-pixel comparison, the semantic information contained in the representations produced by the model is compared. This drives the optimization of the weights to capture the contextual and structural content of the image, rather than faithfully reconstructing every detail at the pixel level, as is the case in Generative Architectures.

One problem with this type of model is the possibility of collapse, that is, the risk that the model will learn to produce the same constant embedding regardless of the input, thereby "cheating" the loss function, as all embeddings automatically become similar to one another (energy always low), without any useful information having been learned. It is precisely on the basis of the strategy adopted to prevent collapse that these models are divided into two broad categories. *Contrastive* methods, such as *SimCLR* [19], prevent collapse by explicitly comparing positive pairs (views of the same image) with a large set of negative pairs (views of different images), typically sampled within the same batch; this requires batches of considerable size to ensure a sufficient number of negative examples. *Non-contrastive* methods, including *SimSiam* [20] — used in this thesis — prevent collapse without resorting to negative pairs, by

adopting architectural and procedural measures at the training level, described in detail in the following section. Generative Architectures, by contrast, do not suffer from this type of problem.

Joint-Embedding Predictive Architectures This family of models, of which *I-JEPA* [21] is a part, is conceptually similar to Generative Architectures: starting from a signal x , conditioned on a latent variable z (the mask), the model learns to predict a signal y . The key difference lies in the space in which the loss is computed: not in the input space, as in Generative Architectures, but in the embedding space. Unlike Joint-Embedding Architectures, however, the aim is not to obtain representations that are invariant with respect to a manually selected set of data augmentation techniques. No data augmentation is used here: rather, the model learns to predict the embedding of one portion of the signal based on the embedding of another portion, conditioning the prediction on the additional information z . This makes the learning of the representation independent of the choice of augmentations, making the model more robust but also more complex. However, this family of models is also subject to the problem of representational collapse, just like Joint-Embedding Architectures.

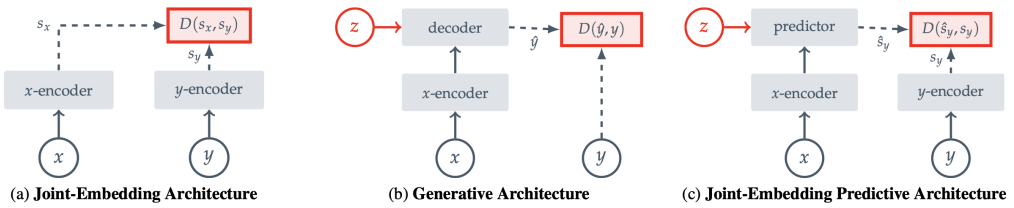


Figure 1: A comparison of the main paradigms of self-supervised learning [21].

2.2.3 SimSiam: a Non-Contrastive Joint-Embedding Architecture

Specifically, this part of the project employs a non-contrastive, self-supervised approach based on Joint-Embedding Architectures, known as SimSiam [20]. In this type of training, ResNet50 [22] is used as the encoder; this is a deep CNN designed to overcome some of the typical limitations of classical deep

convolutional networks, such as the vanishing gradient problem and optimization difficulties, through the introduction of so-called *skip connections*.

This convolutional network was therefore combined with SimSiam training, in which the supervision signal is not provided by external labels or by comparison with negative examples, but emerges intrinsically from the comparison between different views of the same image.

The architecture of this training method (Fig. 2) is organized according to a Siamese paradigm, consisting of two symmetrical branches that share the same weights. Both branches use the same encoder $\mathbf{f}$, comprising a ResNet50 backbone followed by a projection MLP. Starting from a single input image $\mathbf{x}$, two distinct views $\mathbf{x}_1$ and $\mathbf{x}_2$ are generated by applying random data augmentation transformations. The two views are processed in parallel by the two branches of the encoder, generating two representation vectors. Subsequently, only one of the two vectors is passed through an additional module, called predictor $\mathbf{h}$, while the other is treated as a fixed target. The objective of the training is to minimize the distance (i.e. maximize the similarity) between the obtained representations: although the two views $\mathbf{x}_1$ and $\mathbf{x}_2$ appear visually different due to the transformations applied, they derive from the same image and therefore share the same semantic information.

As discussed earlier, SimSiam is also potentially subject to the problem of representational collapse: to prevent this, a *stop-gradient* operation is applied to the branch that does not pass through the predictor, which halts the propagation of the gradient along that path, effectively making that representation a fixed target against which the other branch is optimized. The authors of SimSiam interpret this mechanism as an implicit form of the *Expectation-Maximization* algorithm: the branch isolated from the gradient performs a function analogous to the *E step*, providing a stable target, while the predictor, trained to bridge the gap relative to that target, performs a function analogous to the *M step*. It is precisely this dual asymmetry — architectural, due to the presence of the predictor on only one branch, and in the gradient flow, due to the *stop-gradient* on the other — that prevents the two branches from simply collapsing towards the same constant representation, even in the

complete absence of negative pairs or explicit regularization mechanisms. To empirically verify the absence of the phenomenon of representation collapse during training, one can monitor the standard deviation of the embeddings generated from distinct images.

One final crucial aspect to consider is the heavy reliance of this type of training on the choice of data augmentation techniques used. If the applied transformations are too mild, the task becomes trivial because the two augmented views are nearly identical. Conversely, overly aggressive transformations make the task too difficult, hindering the model’s convergence. Therefore, identifying a suitably balanced set of augmentations is one of the most delicate aspects of the entire training process, particularly in the case of electron microscopy images, where certain standard transformations cannot be applied to avoid excessively distorting the biological information contained in the images. The details of these aspects will be discussed in the *Material and Methods* section.

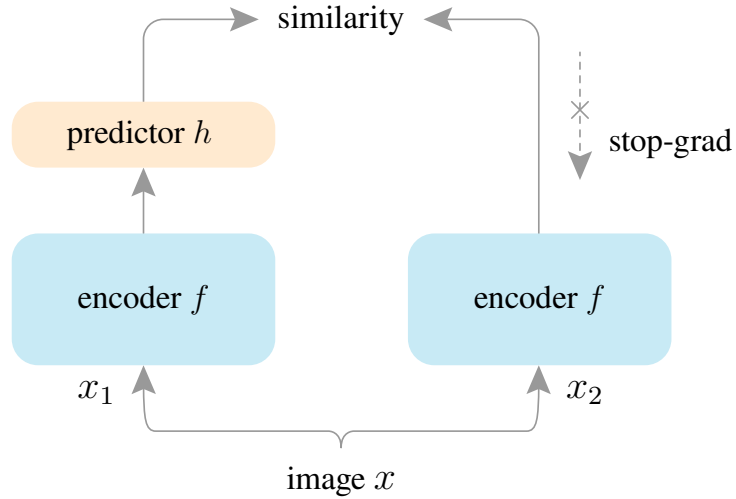


Figure 2: SimSiam architecture [20]

3 Material and Methods

This section will describe in detail the data used, the pre-processing procedures, the model architectures and training, along with the relevant hyperparameters and downstream analyses. To make it easier to read, the section has been divided into two main parts, corresponding to the two projects carried out as part of this thesis. The discussion and interpretation of the results will, however, be presented in the following section, *Results and Discussion*.

3.1 Supervised Learning: Mitochondria Segmentation

3.1.1 Dataset and Data Pre-Processing

Two different data sources were used for mitochondrial segmentation. The first dataset consists of 52 pairs of images and their corresponding labels, kindly provided by Prof. L. Leanza and Prof. N. Plotegher, both of the Department of Biology at the University of Padua. The images, acquired using TEM technology in grayscale, have a digital resolution of 2048×2048 pixels. The original labels, provided in JSON format, included annotations for various types of organelles (mitochondria, endoplasmic reticulum, nucleus, etc.), each identified by a sequential number. To isolate our region of interest, a threshold was applied to extract only the masks relating to mitochondria: pixels belonging to the mitochondrial class were assigned a value of 1, while all remaining pixels—including those originally annotated as belonging to other organelles—were set to 0, thereby generating the required binary labels. Finally, to ensure compatibility with the input requirements of the model, the original grayscale TEM images were converted to a 3-channel RGB format. Following this preprocessing pipeline—and the removal of one pair due to a missing annotation—the resulting dataset comprised a total of 51 valid image-mask pairs.

The second dataset, on the other hand, was sourced from a public online repository¹ and consists of 27 pairs of images and labels relating to striated muscle

¹<https://drive.google.com/drive/folders/1n2ZqbJEHPyMB6a60TBBACt5Jct2PZJc?usp=sharing>

tissue. This tissue is particularly rich in mitochondria, which are larger in size than those in the previous dataset, and the images have a digital resolution of 2560×2560 pixels. Unlike the first dataset, these images were already provided in 3-channel RGB format and therefore did not require any channel conversion. Otherwise the labels were provided as 3-channel RGB images; therefore, a conversion to the binary pseudocolor format was required, assigning a value of 0 to background pixels and a value of 1 to mitochondrial pixels.

As the two sources contained images of different resolutions (2048×2048 and 2560×2560 pixels, both originally square in format) and the model requires images of uniform size as input, all images were resized to the common target size of 1024×1024 pixels, thereby also reducing the computational load required by the model.

By combining the two datasets obtained in this way, a single dataset of 78 image-label pairs was constructed, divided into an 80% training set and a 20% validation set. The training set was further divided into six sub-datasets of increasing size, in increments of 10 images (10, 20, 30, 40, 50, 62), while the validation set remained unchanged for each configuration, containing 16 image-label pairs. The sub-datasets were constructed incrementally and cumulatively: each sub-dataset fully includes the images from the previous one, with the addition of 10 new images until the target size is reached (for example, the 30-image sub-dataset contains the 20 images from the previous sub-dataset plus 10 new images, and so on). This division allowed us to progressively train a U-Net on an increasing number of images, with the aim of analyzing the impact of the amount of training data on the model’s performance in the segmentation task. Each of the six sub-datasets was finally organized according to the structure required by the PaddleSeg framework [23]. The figure 3 shows an example of image-label pairs from each of the two datasets.

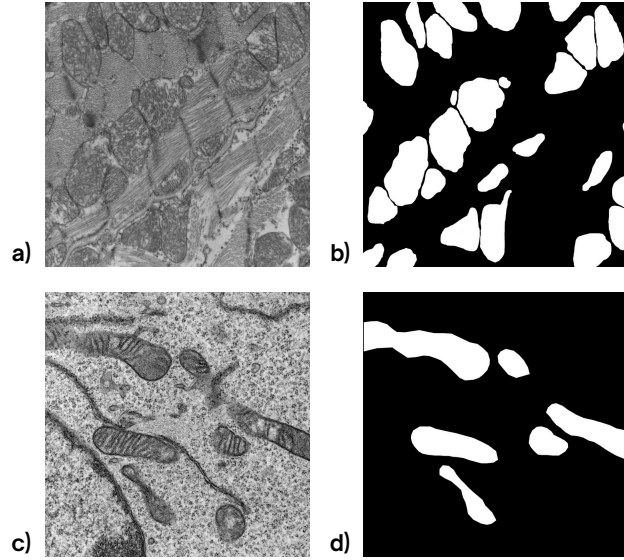


Figure 3: Examples of TEM images associated with their respective mitochondrial annotations. (a) Image from the striated muscle tissue and (b) the corresponding ground truth label. (c) TEM image courtesy of Prof. L. Leanza and Prof. N. Plotegher, shown alongside (d) its corresponding segmentation mask.

3.1.2 Data Augmentation

During the model training phase, a data augmentation pipeline was applied in order to increase the variability of the data —given that the dataset was small— and to improve the model’s generalization ability. Specifically, a *ResizeStepScaling* transformation was applied to each image, with a random scaling factor between 0.5 and 2.0 in increments of 0.25, followed by a *RandomPaddingCrop* of size 512×512 pixels. This combination simulates acquisitions at different effective scales: depending on the applied scaling factor, the subsequent crop selects larger or smaller portions of the image, automatically adding padding if the scaled image is smaller than the target size. Furthermore, geometric transformations such as *RandomHorizontalFlip* and *RandomVerticalFlip* were applied with a probability of 0.5 to simulate different possible acquisition orientations.

To introduce photometric variability, the *RandomDistort* transformation was used to perturb brightness and contrast within a range of $\pm 40\%$ relative to the original value. However, the saturation component —which is enabled by default in this transformation— was disabled. Since the images are in grayscale,

altering saturation would risk introducing biologically irrelevant chromatic artifacts that could confuse the model. Finally, the pixel values were normalized to the $[-1, 1]$ interval (with mean and standard deviation both set to $[0.5, 0.5, 0.5]$) to facilitate the convergence of the optimization process. During the validation phase, only the normalization step was applied, without any data augmentation transformations.

3.1.3 Model Architecture and Training Configuration

The U-Net architecture used in this work follows the standard implementation provided by the PaddleSeg framework [23], which is almost entirely faithful to the structure proposed in the seminal paper [8]. As defined previously, it is organized into a contracting path (encoder) and an expansive path (decoder), connected to each other via skip connections.

In the encoder, the input image (3-channel RGB) is initially processed by a double convolutional block (3×3 convolution, batch normalization, *ReLU* activation), which increases the number of channels from 3 to 64 while maintaining the spatial resolution. This is followed by four downsampling stages, each consisting of a *max-pooling* layer (2×2 kernel, stride 2) that halves the spatial resolution, followed by two convolutional blocks (*Conv-BN-ReLU*) that progressively double the number of channels. At the end of the encoder, the input image is represented by a 512-channel feature map (and not 1024 as in the original implementation) with a spatial resolution dependent on the size of the input image. Before each max-pooling operation, the corresponding feature map is saved and then passed to the decoder via a skip connection. This mechanism allows high-resolution spatial information to be preserved.

The decoder consists of four upsampling stages, mirroring those of the encoder while integrating the information derived from the skip connections. At the end of the four upsampling stages, the feature map returns to the original spatial resolution of the input image, with 64 channels. A final convolutional layer (3×3 kernel, not 1×1 as in the original paper) projects these 64 channels onto the desired number of classes (two in this case), producing the binary segmentation mask in which each pixel is classified as belonging to either the

mitochondrion or *background* class.

As previously described, this model was trained six times using an increasing number of images. Each training run was configured via a customized version of the *drive.yml* configuration file, incorporating the modifications detailed both above and below, and executed following the official documentation provided by PaddleSeg.

Due to the significant difference in size between the various data subsets, each training phase was conducted by keeping the number of epochs fixed at 300 and setting the *batch size* to 8. This choice allowed us to standardize the experimental conditions, making the results more comparable across the different runs.

For all training runs, a *Stochastic Gradient Descent* (SGD) optimizer was used, with an initial learning rate of 0.01, a polynomial decay scheduler (power 0.9), momentum of 0.9 and weight decay of 4.0×10^{-5} . A *Dice Loss* function was adopted, with class-specific weights (0.2 for the background, 0.8 for the mitochondrial class), to compensate for the class imbalance.

For each training run, various performance metrics were computed by comparing the model’s predictions on the validation set with the corresponding ground-truth binary masks. In particular, metrics such as mIoU and single-class IoU were calculated, both of which are particularly well suited to tasks involving image data. The IoU is calculated individually for each of the classes present in the image (background and mitochondrion) as the ratio between the area of overlap and the area of union between the prediction and the ground truth. This training configuration was used for each of the six sub-datasets described in section *Dataset and Data Pre-Processing*. The results of this step will be discussed in the *Results and Discussion* section.

3.2 Self-Supervised

3.2.1 Dataset and Data Pre-Processing

For the self-supervised part of this thesis, a total of five datasets of histological and ultrastructural images were considered. These datasets were used both for preliminary self-supervised pre-training experiments (combining them in various ways) and for subsequent analyses focused solely on the main dataset relating to GD. The main dataset, hereinafter referred to as the *Gaucher dataset*, consists of images acquired via transmission electron microscopy of tissue sections, taken from patients with GD and healthy control subjects. In total, the dataset comprises 10 distinct samples: 3 control subjects (healthy) and 7 pathological subjects, the latter characterized by different genotypes and degrees of clinical severity: 3 samples of severity *G1*, 3 of severity *G2*, and 1 of severity *G3* (Table 1). The complete dataset (provided by Prof. N. Plotegher of the Department of Biology at the University of Padua) consists of **3,228** images with an original resolution of 2048×2115 pixels. As the lower portion of each image contains a black band displaying the scale in micrometers—which is not relevant for the purposes of the model—all images were cropped to a square format of 2048×2048 pixels. Each image was then divided into 64 non-overlapping patches of size 256×256 pixels, yielding a total of **206,592** patches.

For classification purposes, the 10 original classes associated with the individual samples were grouped into three macro-classes: *Healthy* (3 control subjects), *G1_G3* (patients with phenotypes 1 and 3), and *G2* (exclusively type 2 patients). This grouping choice, rather than treating the four original phenotypic groups as independent classes, was driven by two main considerations. First, *G1* and *G3* patients share a more similar clinical course, whereas the *G2* variant is characterized by an acute form with very early childhood onset, as described in detail in the *Introduction*; given its markedly higher severity, it was essential to keep *G2* as a distinct class rather than merging it with the others. Second, the original *G3* class was severely underrepresented, comprising only 245 images compared to approximately 1000 images each for *G1* and

$G2$; training a classifier on such an imbalanced fourth class would have made it impossible to obtain a reliable classification. Merging $G3$ with the clinically similar $G1$ class therefore also served to mitigate this class imbalance. This reorganization was further necessary to prevent the linear classifier (detailed in the following section) from learning to recognize individual patients. By grouping subjects with a similar clinical-genetic profile, the model was instead forced to discriminate solely on the basis of the actual severity of the condition. For the preliminary multi-dataset pre-training phase alone, a further four datasets were integrated, with the aim of broadening the variability of the visual signal:

- **Dataset 2:** Provided by Prof. N. Plotegher of the Department of Biology at the University of Padua, this dataset comprises **2,731** images of various tissues (colon, duodenum, and cerebellum) obtained from both healthy controls and patients affected by diverse pathologies. Colon and duodenum samples derive from wild-type (WT), heterozygous, and carrier mouse models of the human N370S GBA1 transgene, representing established murine models of GD. Cerebellum samples, on the other hand, consist of primary cerebellar granule neuron cultures obtained from mice treated with a GBA2 enzyme inhibitor, compared against untreated controls; mutations in GBA2 are known to cause a rare form of hereditary spastic paraplegia. Following the same pre-processing steps applied to the Gaucher dataset (removal of the scale bar, deduplication, and tiling), it yielded **174,784** patches measuring 256×256 pixels.
- **Dataset 3 (LRRK2):** The dataset consists of **376** images organized into 6 groups, corresponding to 3 control subjects and 3 patients with the condition. Specifically, the images compare subjects who are carriers and non-carriers of the mutation in the *LRRK2* gene, which is closely associated with Parkinson’s disease. The division into patches yielded **24,064** images measuring 256×256 pixels. This dataset was also provided by Prof. N. Plotegher of the Department of Biology at the University of Padua.

- **Dataset 4:** comprising **100** TEM images, downloaded from the *EM-PIAR* repository ², of varying sizes and from different tissues and published works. To standardise them, a letterboxing technique was applied at 1024×1024 pixels, subsequently yielding **1,600** patches of 256×256 pixels.
- **Dataset 5:** consisting of **51** unlabeled images, from which **3,264** patches of 256×256 pixels were derived. Notably, this is the same dataset utilized during the mitochondrial segmentation phase (as described in Section 3.2.1), but employed here without the corresponding labels.

Overall, combining all the datasets resulted in a corpus of **6,486** images, from which a total of **410,304** patches were extracted. It should be noted that for all the training processes described below, only 256×256 pixel patches were used, and not the entire high-resolution images.

Grade GD	Macro-class	Mutation	Class Name	N° Images
1	G1_G3	84GN370S	disease_G1_84GN370S	230
1	G1_G3	N370S/N370S	disease_G1_N370SN370S	521
1	G1_G3	N370S/W184R	disease_G1_N370SW184R	230
3	G1_G3	L444P/L444P	disease_G3_L444PL444P	245
2	G2	L444P/L444P	disease_G2_L444PL444P	528
2	G2	C342G/G325R	disease_G2_C342GG325R	301
2	G2	L444P/P415R	disease_G2_L444PP415R	204
–	Healthy	–	healthy_001	532
–	Healthy	–	healthy_002	263
–	Healthy	–	healthy_003	174
Total				3228

Table 1: Distribution of TEM images across GD severity grades, classification macro-classes, associated genetic mutations, and original sample classes, along with the corresponding number of images per group.

3.2.2 Data Augmentation

During SimSiam training, a crucial factor in ensuring that the ResNet-50 backbone effectively condenses the structural information of the images into a meaningful representative vector (embedding) is the adoption of a properly tailored data augmentation pipeline. The aim is to apply transformations that

²<https://www.ebi.ac.uk/empiar/EMPIAR-10982/#>

do not excessively distort the biological information within the TEM images, carefully avoiding both the introduction of artificial artifacts not normally found in the tissue, and the removal of essential structural details inherently present in the samples. In particular, a data augmentation strategy comprising the following transformations was defined for the training set:

- *RandomResizedCrop* with a target size of 256×256 pixels at a scale between 0.6 and 1.0 (probability = 1.0).
- *RandomHorizontalFlip* and *RandomVerticalFlip*, both with a probability of 0.5, to account for the fact that organelles may appear with different orientations within the cell.
- *ColorJitter*, applied by varying only *brightness* and *contrast*, each with a probability of 0.8. Both the *brightness* and *contrast* intensity factors are set to $v = 0.4$, ensuring that the randomly selected values fall within the range $[\max(0, 1 - v), 1 + v]$. This augmentation simulates the variability in *brightness* and *contrast* found in images acquired using different electron microscopes; indeed, different instruments may produce images with slightly different exposure levels. The *saturation* component, however, was omitted, as it would introduce biologically irrelevant chromatic alterations, risking confusing the model.
- *GaussianBlur*, with a radius (σ) sampled from a uniform distribution in the interval $[0.1, 1.4]$, applied with a probability of 0.5. This transformation simulates the noise and slight focus variations normally present in TEM images.
- Finally, the image was converted to a tensor and subsequently normalized using the mean and standard deviation calculated from the entire dataset used during training.

As can be seen, each of these transformations (with the exception of the conversion to a tensor and the normalization) was applied randomly to the input

images. In this way, at each training step, the two views generated from the same image provide distinct representations of the original information; this is essential for the learning paradigm underlying SimSiam to be effective.

3.2.3 Training Configuration

For the self-supervised training, the ResNet-50 architecture implemented in PyTorch [24] was therefore selected—specifically, the exact configuration utilized by the original authors of SimSiam. Downstream of the backbone is a projection MLP (forming the last part of the encoder f), consisting of three fully-connected layers, each followed by *Batch Normalization*. The hidden layers, each with a dimension of 2048, use the *ReLU* activation function, while the output layer also has a dimension of 2048 and does not use any activation function. This is followed by a prediction MLP (h), consisting of two layers: the hidden layer is followed by *Batch Normalization* and *ReLU*, while the output layer has neither *Batch Normalization* nor *ReLU*. The final layer of h has a dimension of 2048, while its hidden layer has a dimension of 512, creating a bottleneck structure.

As mentioned in the introduction, for each input image x , two different data augmentations are created (Sec. 3.2.2), thereby generating two different representations, x_1 and x_2 , of the same image. Both views are processed by the shared encoder f , yielding two representations $z_1 = f(x_1)$ and $z_2 = f(x_2)$. Only one of the two branches then applies a further module, the predictor h (a second MLP), producing a prediction $p_1 = h(z_1)$; the other branch, however, is not further transformed and its output z_2 is treated as the target, isolating it from the gradient calculation via a *stop-gradient* operation to prevent collapse. The negative cosine similarity (Eq. 1) is calculated between the prediction p_1 and the target z_2 (similarly, between p_2 and z_1 , swapping the roles of the two branches); this is a measure of directional distance between the two vectors. By combining the two terms calculated in the two directions, we obtain a symmetric loss function (Eq. 2), lying in the interval $[-1, +1]$, which guides the updating of the model’s weights via backpropagation.

$$D(p, z) = -\frac{p}{\|p\|_2} \cdot \frac{z}{\|z\|_2} \quad (1)$$

$$L(p, z) = \frac{1}{2} \cdot D(p_1, \text{stopgrad}(z_2)) + \frac{1}{2} \cdot D(p_2, \text{stopgrad}(z_1)) \quad (2)$$

For the training of this architecture, the same hyperparameter configuration proposed by the original authors of SimSiam was adopted. SGD was used as the optimizer, with the learning rate scaled linearly according to the batch size (Eq. 3), with a base value of $lr = 0.05$; the learning rate follows a cosine scheduler. A weight decay of 0.0001 and a momentum of 0.9 were also set, with a batch size of 128. SimSiam was trained for 300 epochs. In addition to the loss curve, the standard deviation of the embeddings was monitored at each epoch to verify the stability of the training and prevent collapse. In theory, the standard deviation should remain at an ideal value of $1/\sqrt{d}$ [20], where d represents the dimension of the embedding vector; 2048 in our case.

This training configuration is applied in three distinct experimental scenarios. First, the model is trained on the entire collection of datasets to obtain a general-purpose encoder, intended for a subsequent fine-tuning step.

Second, to qualitatively evaluate the representations learned by the self-supervised model, training is performed exclusively on the complete Gaucher dataset (comprising all ten patients); this allows for exploratory analyses of the embedding space, such as UMAP visualizations.

Finally, to quantitatively assess the quality of the generated embeddings and the model’s generalization capabilities, the network is trained on the Gaucher dataset using nine out of the ten available patients. This leave-one-patient-out (LOPO) approach enables two distinct assessments: a quantitative linear evaluation, used to test embedding quality and model generalization on an unseen subject, and a SpotSweeper-like analysis, designed to evaluate performance exclusively on the most informative patches.

The rationale and outcomes of these three approaches will be discussed in detail in the *Results and Discussion* chapter.

3.2.4 Downstream Analysis

This section presents the analyses conducted following the pre-training of the network using the SimSiam architecture. These evaluations are necessary because self-supervised learning is not an end in itself, but rather aims to generate embeddings for use in subsequent downstream tasks. To this end, two different quantitative evaluations were performed, building respectively on the encoder pre-trained on the entire collection of datasets and on the encoder pre-trained on the Gaucher dataset excluding one patient; the third pre-training scenario, involving the complete Gaucher dataset, is instead reserved for a purely qualitative analysis of the embedding space, as detailed in the previous section.

The first involves *fine-tuning* the model on the Gaucher dataset, starting from this general-purpose encoder. The aim is to specialize the general representative features learned during pre-training to the specific dataset of interest.

The second evaluation consists of a *linear evaluation*, used in conjunction with subsequent analyses to assess the quality of the embeddings. These embeddings were obtained by training the ResNet-50 via self-supervised learning exclusively on the Gaucher dataset. In this case, 9 of the 10 available patients were used, while the remaining patient was excluded from the training phase and reserved as a test set to assess the model’s generalization ability on a previously unseen subject. We also examined how performance changes when using only the informative patches, identified via the *SpotSweeper-like Analysis*, thereby excluding non-informative patches from the study. In both cases, the output layer of the linear classifier was configured with three neurons, corresponding to the three target classes described in Section *Dataset and Data Pre-Processing*

Fine-Tuning As mentioned, this analysis was conducted using the encoder that had been pre-trained via self-supervised learning on the entire available dataset, followed by the joint retraining of the backbone and the linear classifier exclusively on the Gaucher dataset, thereby tailoring the model specifically to the latter. To this end, the final checkpoint from the self-supervised training was retrieved, and only the ResNet-50 parameters were extracted, excluding

those relating to the projector and predictor. A single-layer linear classifier with three neurons in the output layer was then attached to the encoder to classify each image into its respective subclasses within the Gaucher dataset. To prevent catastrophic forgetting from affecting the encoder, the backbone was kept frozen for the first 20 epochs, with only the linear head being trained. During the early stages of training, the gradients calculated from the randomly initialized classification head are of high magnitude; propagating these updates through the entire network would risk degrading the visual representations learned during the self-supervised phase. During this initial interval (epochs 1–20), optimization focused exclusively on the linear head, following the configuration proposed in the original SimSiam paper: a *Stochastic Gradient Descent* optimizer was used, with momentum set to 0.9 and weight decay set to *zero*. The base learning rate was set to 30 and scaled linearly according to the batch size (128), as shown in Equation 3:

$$lr = \text{base_lr} \times \frac{\text{batch_size}}{256} \quad (3)$$

This value was kept constant throughout the whole training interval of the classifier. The weights of the linear head were sampled from a normal distribution $N(0, 0.01)$, while the biases were initialized to *zero*. From the 21st epoch onward, the ResNet-50 encoder was unfrozen to allow for end-to-end fine-tuning of the entire architecture. In this second phase, the encoder was optimized with an initial learning rate of 0.01 (also scaled linearly, as in Equation 3), a momentum of 0.9, and *zero* weight decay. The modulation of the encoder’s learning rate over the following epochs was managed using a *cosine decay scheduler*. To make the training more robust, data augmentation was employed, albeit to a lesser extent than that used in pre-training. Specifically, the following transformations were applied to the training set: *RandomResizedCrop* (size 256×256 , scale between 0.6 and 1.0), *RandomHorizontalFlip*, and *RandomVerticalFlip* (both with a probability of 0.5), followed by tensor conversion and data normalization consistent with that used in SimSiam. For the validation dataset, the pipeline was limited to tensor conversion and nor-

malization only, in order to avoid alterations that are not representative of actual performance. Fine-tuning was carried out for a total of 150 epochs with a batch size of 128. At each epoch, the loss, accuracy, and Matthews Correlation Coefficient (MCC) were monitored; the MCC was specifically chosen as it accounts for class imbalance mathematically. Additionally, for this evaluation, the original dataset was split into a training set and a validation set, comprising 80% and 20% of the total data, respectively.

Linear Evaluation Following the pre-training of the ResNet-50 encoder using SimSiam on the Gaucher dataset—from which one patient was excluded a priori and set aside as a test set to evaluate the model’s generalization ability on a subject never observed during training—a linear evaluation was conducted to quantitatively measure the quality of the extracted representations. This was done without modifying the encoder’s weights, so as to assess solely the quality of the parameters obtained through self-supervised training.

Similar to the fine-tuning phase, the checkpoint from the SimSiam training on the single dataset was retrieved, excluding the projector and the predictor, and a single-layer linear classifier with an output layer consisting of three neurons was attached to the encoder. During this procedure, the encoder’s parameters were kept strictly frozen, with only the linear classifier being optimized.

The same data augmentation transformations used in fine-tuning were adopted for both training and validation. The classifier’s hyperparameters were kept unchanged, with the exception of the learning rate, which was modulated using a *cosine decay scheduler*: the high initial value encourages exploration of the parameter space in the early epochs, while the gradual reduction allows for stable convergence towards the minimum of the loss. For this phase, the remaining nine patients were split into an 80% training set and a 20% validation set. The classifier was then trained for 120 epochs with a batch size of 128, while monitoring the evolution of the training loss, the validation loss, the accuracy, and the MCC. Subsequently, using the checkpoint of the best model, classification was performed on the held-out patient to assess the generalization of the resulting model.

SpotSweeper-like Analysis Following the Linear Evaluation, the SpotSweeper-like analysis [25] was implemented; this is a quality control (QC) method originally developed for the analysis of spatial transcriptomic data [26], a technique that allows the measurement of gene expression — defined as the amount of RNA transcribed by a given gene — at each point (*spot*) in a spatial grid, thereby integrating information on gene expression with its location within the tissue. Unlike many other QC methods, which evaluate each spot in the grid in isolation or by considering all other spots collectively, SpotSweeper estimates spot quality by taking into account only a local neighborhood, comprising the k nearest neighbors, thereby enabling the identification, through successive comparisons, of which regions of the grid are of high or low quality, and discarding those of poor quality in order to reduce the risk of bias in subsequent analyses.

TEM images similarly contain numerous non-informative or low-quality regions (structure-free cytoplasm, artifacts, bubbles, etc.), while only a limited proportion of regions are actually relevant for classification purposes. In this work, the same principle underlying SpotSweeper was therefore adapted to patches extracted from TEM images, with the aim of distinguishing informative patches from non-informative ones, thereby identifying the regions of the image that are actually useful for the classification task.

To this end, the encoder with the linear classifier previously trained on the 9 patients in the Gaucher dataset was used to perform inference on the entire dataset, without distinguishing between the training and validation sets, thereby obtaining the corresponding embedding for each patch. For each embedding, rather than considering the class predicted by the model, the probability assigned by the classifier to the patch’s true class was extracted, thus providing a measure of how ”convincingly” the model associates each input with its correct class. This yields, for each patch, an embedding–probability pair, which is used as input for the SpotSweeper-like analysis.

For each embedding, the k nearest neighbors in embedding space were subsequently identified using the *Nearest Neighbors algorithm* implemented in the *FAISS* (Facebook AI Similarity Search) library [27], yielding a matrix in

which, for each vector, the indices of the k vectors closest to it are listed. The similarity between embeddings is calculated using *Euclidean distance*. The analysis was carried out for different values of $k \in \{6, 18, 36, 60, 90\}$, representing neighborhoods of increasing size. For each value of k and for each embedding, the probabilities associated with the true class of its k nearest neighbors were considered, and their mean and variance were calculated. The result of this procedure is two matrices — one for the mean, one for the variance — in which the rows correspond to the patches/embeddings and the columns to the different values of k ; each cell therefore contains the mean (or variance) of the probability associated with the true class, calculated over the neighborhood of size k . In this way, we can see, on average, how correctly that neighborhood was classified by the model.

PCA was applied to the matrix of mean profiles (and similarly to that of variance profiles) to reduce the multi-scale profile defined by the different values of k to just two principal components, thereby condensing the trend in model confidence from the micro-neighborhood ($k = 6$) to the macro-neighborhood ($k = 90$). Finally, *K-means* clustering with two clusters was performed on the resulting two principal components, with the aim of separating the patches into two distinct clusters — *informative* and *non-informative*. The points were then plotted in the space of the two principal components, colored according to their actual labels, and for each of the two resulting clusters, the overall accuracy was calculated by comparing, for each vector in the cluster, the labels with the predictions.

3.3 Experimental setup

All the analyses carried out in this thesis were implemented in *Python*. The PaddleSeg framework was used to prepare the dataset and train the U-Net for mitochondrial segmentation, following the tutorials on the website in full and using the U-Net architecture available in the repository. Comprehensive tutorials on how to create the dataset, train the model, and perform related tasks can be found in the "*Tutorials*" section of the official PaddleSeg GitHub repository [23].

The SimSiam self-supervised pipeline (based on the ResNet50 backbone) was developed by integrating and adapting the official implementation provided by the authors of the original paper. Downstream analyses, dimensionality reduction and data visualization were carried out using specialized libraries for machine learning and image processing, including *PyTorch*, *torchvision*, *PyTorch Lightning*, *Scikit-learn*, *FAISS*, *Pandas*, *NumPy*, *Pillow*, *Bokeh*, *Plotly*, *Matplotlib* and *Seaborn*.

To ensure the traceability and full reproducibility of the work, the entire source code was managed using the *Git* version control system, while the software environment and its dependencies—including the exact versions of Python and all the packages used—were configured and fixed using the *uv* package manager. The project repository, containing the final version of the code and the *uv.lock* file, can be accessed at the following address: <https://github.com/sales-lab/lysoem.git>.

The entire computational workload was run within a virtual machine based on the KVM hypervisor, utilizing a high-performance *NVIDIA L40S* graphics processing unit equipped with 46 GB of dedicated VRAM, supported by NVIDIA drivers (v610.57.04) with CUDA 13.3 acceleration.

4 Results and Discussion

This section presents and discusses the results obtained from the analyses described above. The discussion begins with an examination of the performance of semantic segmentation based on the supervised U-Net architecture, before moving on to an analysis of the results achieved during the various stages of the self-supervised learning project.

4.1 Supervised Learning: Mitochondria Segmentation

As described above, for semantic-segmentation analysis a single dataset comprising 78 image-mask pairs was used. This was subsequently divided into six training sub-datasets of incremental and cumulative sizes (10, 20, 30, 40, 50 and 62 images) and a validation dataset of fixed size (16 images). A U-Net architecture was then trained for each sub-dataset, and metrics were collected to assess the quality of the training process.

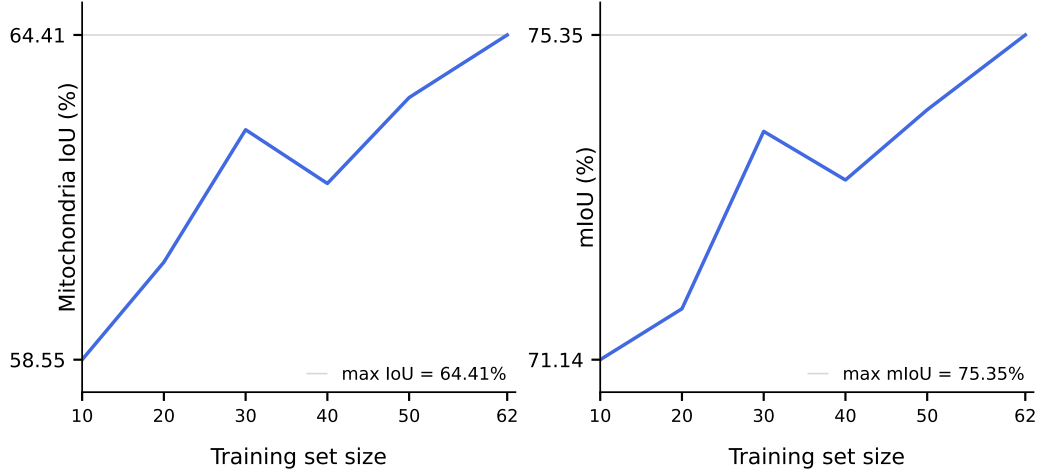


Figure 4: Mitochondria IoU and Mean IoU (mIoU) for the U-Net model across incremental and cumulative training dataset setups. The validation set remains fixed with 16 images.

As the results show, the Intersection over Union achieves satisfactory values, with an mIoU of **75.4%** and a class-specific IoU for the mitochondrial class of approximately **64.4%**. The overall mIoU is higher than the class-specific IoU for the class of interest due to the clear prevalence of background pixels over mitochondrial pixels. The model has managed to achieve a fairly high

level of performance despite using a very limited number of training images: these metrics appear to be constrained not by the limitations of the model itself, but rather by the small size of the training set, suggesting that expanding the sample would lead to a further improvement in performance.

A critical issue, however, lies in the high average quality of the images used: to make the model more robust and generalizable, it would be advisable to incorporate not only a greater volume of data but also greater sample variability, thereby freeing the network from its dependence on high-resolution acquisitions.

Overall, the results confirm the effectiveness of this type of architecture in ensuring automatic, rapid and accurate segmentation. Such a reliable pipeline provides a solid foundation for subsequent advanced morphometric analyses, such as the quantification of mitochondrial area and perimeter, the counting of cristae and the characterization of interaction networks between mitochondria.

4.2 Self-Supervised Learning

This section presents the results of the three self-supervised experiments introduced in the previous chapters (Sec. 3.2.3 and Sec. 3.2.4) — generic pre-training on the entire dataset collection, Gaucher-only pre-training, and Gaucher pre-training with one patient held out — together with their corresponding downstream tasks. For the third experiment, model performance is assessed both on the three classes of interest ($G1_G3$, $G2$ and *Healthy*) and, using a *leave-one-patient-out* (LOPO) strategy, on the excluded patient, to verify whether the model correctly generalizes to a subject never observed during either self-supervised or supervised training.

4.2.1 General pre-training and fine-tuning on the Gaucher dataset

In the first stage, ResNet50 was trained in a self-supervised manner using SimSiam, combining the Gaucher dataset with other TEM image datasets, for a total of 410,304 patches, with the aim of learning a general visual representation from a large volume of unlabeled data. Subsequently, the network was

fine-tuned by applying a linear classifier to the Gaucher dataset alone for 3-class classification.

The results of this configuration proved unsatisfactory, achieving a maximum accuracy of **40%**. This behavior can be attributed to the high heterogeneity of the datasets used in pre-training, whose morphological characteristics were out-of-distribution with respect to the target domain: rather than extracting generalizable structural features, the high diversity of the images prevented the model from learning informative representations, yielding features that were non-specific and unsuitable for the classification task on the Gaucher dataset. In light of these results, the approach based on global pre-training followed by fine-tuning was abandoned. It was therefore decided to focus subsequent investigations solely on the Gaucher dataset, as the other datasets were found to be out-of-distribution and, consequently, unsuitable for facilitating the learning of representations useful for the target task.

4.2.2 Pre-training on Gaucher dataset and qualitative analysis of the latent space

The self-supervised training of the ResNet50 backbone was then repeated using only the Gaucher dataset, across the entire set of 10 available patients. The aim of this phase was not to obtain a classifier directly, but rather to analyze the latent space generated by the model via UMAP projection, in order to qualitatively assess the arrangement of the learned representations and verify whether the model had captured morphological structures consistent with the classes of interest, despite never having seen labels during training.

The ResNet50 architecture was therefore trained using the self-supervised SimSiam approach on the entire Gaucher dataset for a total of 300 epochs. During training, the trend of the loss function and the standard deviation of the representations were monitored in order to verify the quality of learning and prevent any potential collapse (Fig. 5).

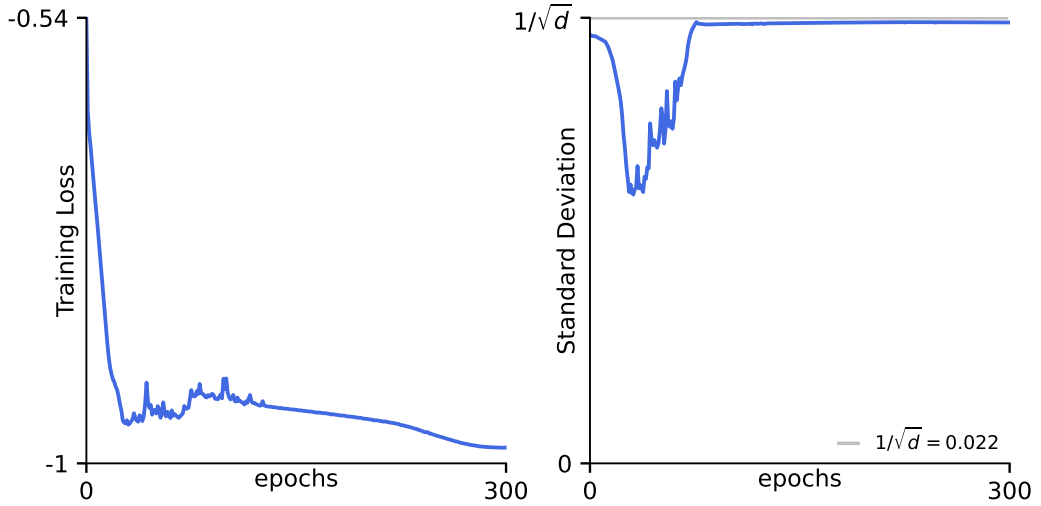


Figure 5: Progression of training loss and standard deviation during SimSiam training.

As the graphs show, the loss function exhibits a downward trend, gradually converging towards the theoretical value of -1 , confirming that the model is learning to represent images more and more effectively within vectors of dimension $d = 2048$. At the same time, the standard deviation stabilizes around the theoretical value $1/\sqrt{d}$, clearly indicating the absence of collapse: the model does indeed generate distinct and informative vectors for different patches. Subsequently, the ResNet50 checkpoint was extracted at the end of training and inference was performed on the same dataset. The embeddings thus obtained were processed using principal component analysis (selecting the first 100 PCs) and visualized using the UMAP dimensionality reduction technique (Fig. 6). In the resulting graph, the points were colored according to the patient to which they belonged.

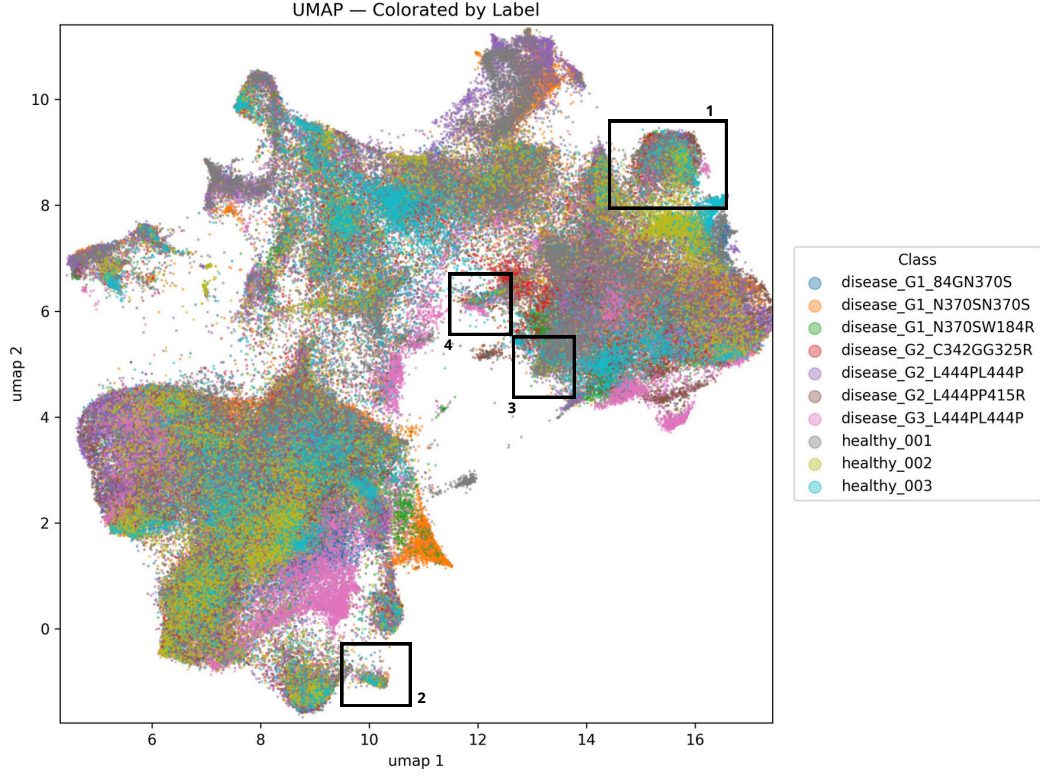


Figure 6: A two-dimensional UMAP visualization calculated using the first 100 principal components, with color coding by patient. The analysis visually illustrates the features extracted by the encoder following SimSiam self-supervised training, highlighting the model’s ability to group similar morphological features without the use of labels.

Analysis of the UMAP plot reveals several well-defined clusters, some of which specifically group data from individual patients. This behavior suggests that the model is capable of capturing differences between the information contained in the various patches, identifying characteristics specific to certain patients.

To investigate this phenomenon further, we visually examined the patches whose vectors are contiguous in UMAP space and therefore belong to the same cluster. Upon zooming in on the various regions highlighted in the plot, it can be seen that these samples represent homogeneous biological structures. Specifically, panel 1 includes exclusively mitochondrial regions (Fig. 7, *panels 1.1 - 1.4*), panel 2 is populated by lysosomal structures (*panels 2.1-2.4*), panel 3 isolates actin filaments (*panels 3.1-3.4*), while panel 4 highlights vesicular structures (*panels 4.1-4.4*), confirming the semantic consistency of the groupings.

This finding provides an initial and important confirmation of the model’s ability to learn visual representations with genuine biological significance. The network has, indeed, demonstrated its ability to distinguish between different subcellular structures even in the presence of similar morphological features, operating in a fully self-supervised manner and relying exclusively on comparisons between the different image augmentations.

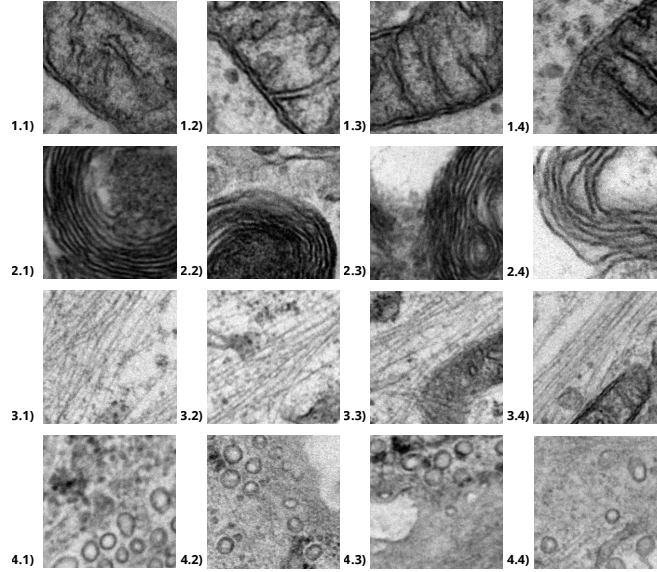


Figure 7: Patches corresponding to contiguous vectors in UMAP space, belonging to the same cluster and characterized by the same biological structure. Each row groups together four patches representative of a specific structure: (1) **mitochondria**, (2) **lysosomes**, (3) **actin filaments**, (4) **vesicles**.

4.2.3 Assessment of generalization using LOPO cross-validation

To rigorously evaluate the network’s ability to generalize to unseen data, a leave-one-patient-out cross-validation strategy was implemented. This approach follows a precise three-stage pipeline: *(i)* self-supervised pre-training of the encoder on the remaining 9 patients; *(ii)* linear evaluation to assess the quality of the learned embeddings; and *(iii)* inference on the held-out patient to accurately measure the model’s true generalization capabilities.

The patient set aside for this purpose is a *G2* subject with the homozygous *L444P/L444P* genotype. The decision to exclude this particular patient is not arbitrary, but reflects the subject’s clinically intermediate position within the

dataset: their genotypic profile lies halfway between that of the other two $G2$ patients (both heterozygous) and that of the homozygous $G3$ patient included in the $G1_G3$ macro-class. This patient therefore represents the ideal candidate for testing the model’s generalization ability on a clinically ambiguous case, situated on the boundary between the classes of interest rather than in a position that is clearly representative of a single category. It should also be noted that, since the training set nonetheless includes two other $G2$ patients (albeit heterozygous), this testing phase does not constitute a blind evaluation of the entire condition, but rather a generalization test on a new individual belonging to a class already known to the model.

Self-supervised pre-training on the 9 patients In the first phase, the ResNet50 model was trained in self-supervised mode on the remaining 9 patients, excluding the entire set of images belonging to the tenth patient from the training. At the end of training, the trends in the training loss and standard deviation were analyzed (Fig. 8); the convergence of the loss towards -1 and the stabilization of the standard deviation at its theoretical peak confirm the absence of the collapse phenomenon.

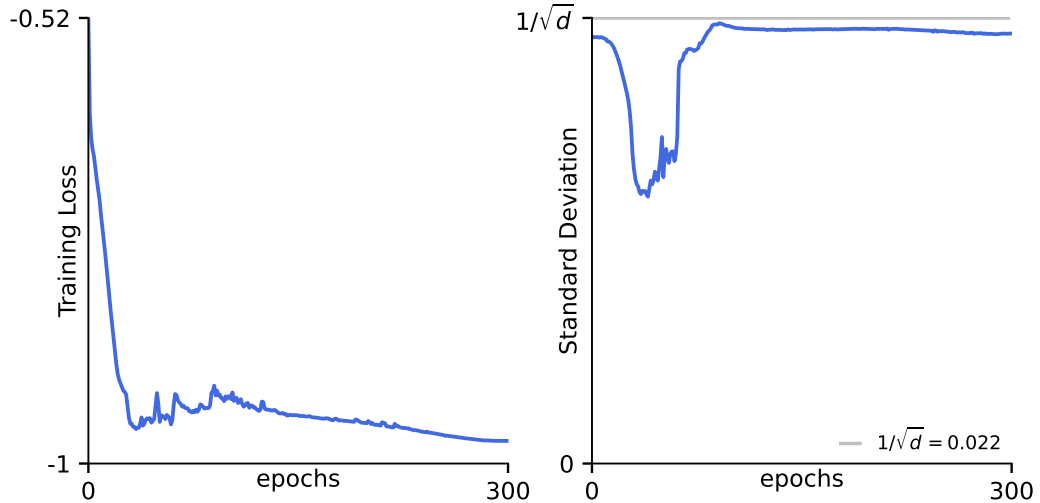


Figure 8: Trend in the *training loss* and *standard deviation* during self-supervised training on the 9 patients in the Gaucher dataset.

Linear Evaluation In the second stage, the quality of the learned embeddings was evaluated via a linear probe. Specifically, a linear classifier was trained and validated on a three-class classification task ($G1_G3$, $G2$, and *Healthy*) using data exclusively from the nine training patients, keeping the held-out patient strictly isolated for the test set. During this phase, the encoder was kept frozen, with only the linear classifier being trained. The decreasing trend in both the training and validation loss curves (provided in the Supplementary Materials) confirms that the optimization process converged properly.

An analysis of the epoch-by-epoch evolution of accuracy and MCC (Fig. 9) reveals that the model reaches a peak accuracy of **75%** and an MCC of **0.60**. The discrepancy between the two metrics is likely attributable to the marked class imbalance in the dataset, a factor that heavily skews accuracy but is inherently accounted for by the MCC, thereby ensuring a more robust evaluation.

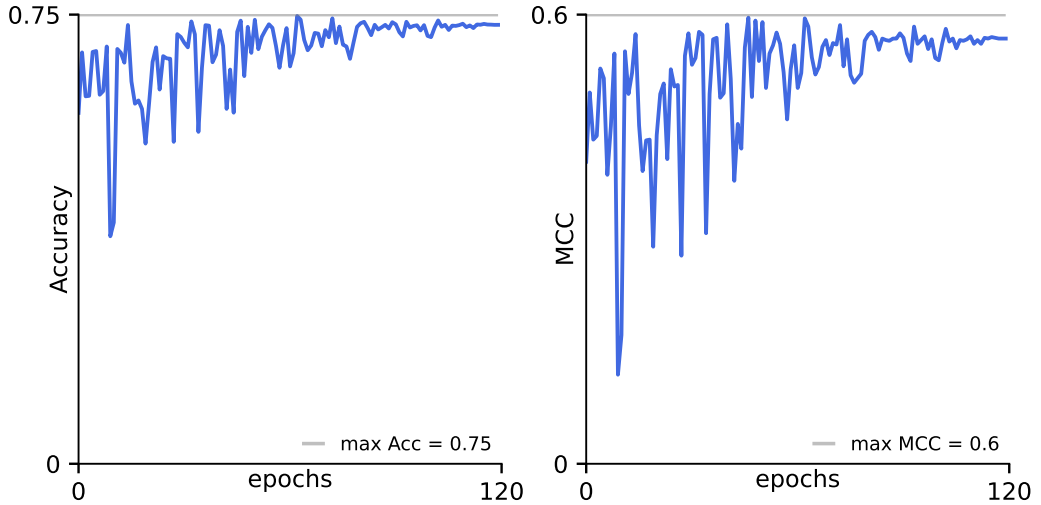


Figure 9: Performance of the linear classifier (*Accuracy* and *MCC*) evaluated on the 9 patients, categorized into the three target classes ($G1_G3$, $G2$, *Healthy*).

This result confirms the network’s ability to extract relevant information from the source images, identifying features capable of effectively distinguishing between classes at the single-patch level. The result obtained is particularly significant when one considers the very nature of TEM images: the majority of the sampled regions are of little diagnostic relevance, consisting of large cyto-

plasmic (and non-cytoplasmic) areas devoid of distinctive biological structures. Only a limited subset of patches contains truly informative morphological features — mitochondria, endoplasmic reticulum, vesicles, lysosomes, membrane segments — that are useful for determining the class to which they belong. While these informative patches contribute positively to the model’s accuracy, the prevalence of non-discriminating regions tends to significantly limit its overall performance.

Evaluation on the held-out patient In the third and final phase, the resulting model—an encoder and a linear classifier, both trained without ever observing the tenth patient— was used to perform inference on the held-out patient. The objective was to measure the extent to which the learned representations could actually generalize to a previously unseen subject. As this patient belongs to a single class (*G2*), it was not possible to calculate the Matthews Correlation Coefficient for this test phase; the evaluation was therefore conducted solely in terms of accuracy.

The accuracy observed on this sample is surprisingly low, standing at approximately **19%**. Specifically, the model misclassifies almost all of the held-out patient’s patches, assigning 41% of the images to the *G1_G3* class and 40% to the *Healthy* class. This marked decline in performance is likely attributable to several factors:

- **Class imbalance:** During the training phase, class *G2* (excluding data from the confidential patient in the test set) is significantly under-represented compared to *G1_G3* and *Healthy*. Specifically, the class with the highest clinical severity has only **32,318** samples, compared with **78,461** for *G1_G3* and **62,016** for *Healthy*. This disparity introduces a systematic bias in the training of the linear classifier, which will tend to favor the discrimination of the majority classes at the expense of the correct identification of the *G2* class.
- **High intra-class heterogeneity:** The condition under investigation,

as described in the *Introduction*, does not exhibit a clear morphological boundary between the different levels of clinical severity. Due to marked inter-patient variability, patches belonging to a *G2* subject may in some cases show greater visual similarity to those in the *G1_G3* or *Healthy* classes than to those of another *G2* subject. This discrepancy may be attributable to the genetic makeup of the samples: while the two *G2* patients included in the training set have a heterozygous genotype, the patient in the test set is homozygous. This genotypic difference could result in substantial phenotypic and ultrastructural variations, hindering the model’s ability to generalize to the test set.

SpotSweeper-like Analysis Following the evaluation of the encoder via the linear classifier, the SpotSweeper-like analysis was applied. This procedure enables the isolation of informative patches from non-informative ones, allowing for a comparative evaluation of the model’s performance on both subsets. Consequently, it becomes possible to observe how the classification metrics change when the assessment is restricted solely to the most relevant patches.

The analysis was conducted on the same dataset used to train the linear classifier (while keeping the tenth patient reserved for the test set). By feeding the images of the nine patients through the network, the corresponding embedding vector was extracted for each patch. For each sample, the mean and variance of the probabilities assigned by the model to the true class were then calculated, evaluated as the number of neighbors k varied.

A PCA was applied to the feature matrix constructed in this way, followed by K-means clustering to separate the data into two distinct groups. The data points were then projected onto the first two principal components and colored according to their cluster to visualize their two-dimensional distribution (Fig. 10). As using only the mean or only the variance produced almost identical distributions, it was decided to continue the analysis considering only the mean of the probabilities.

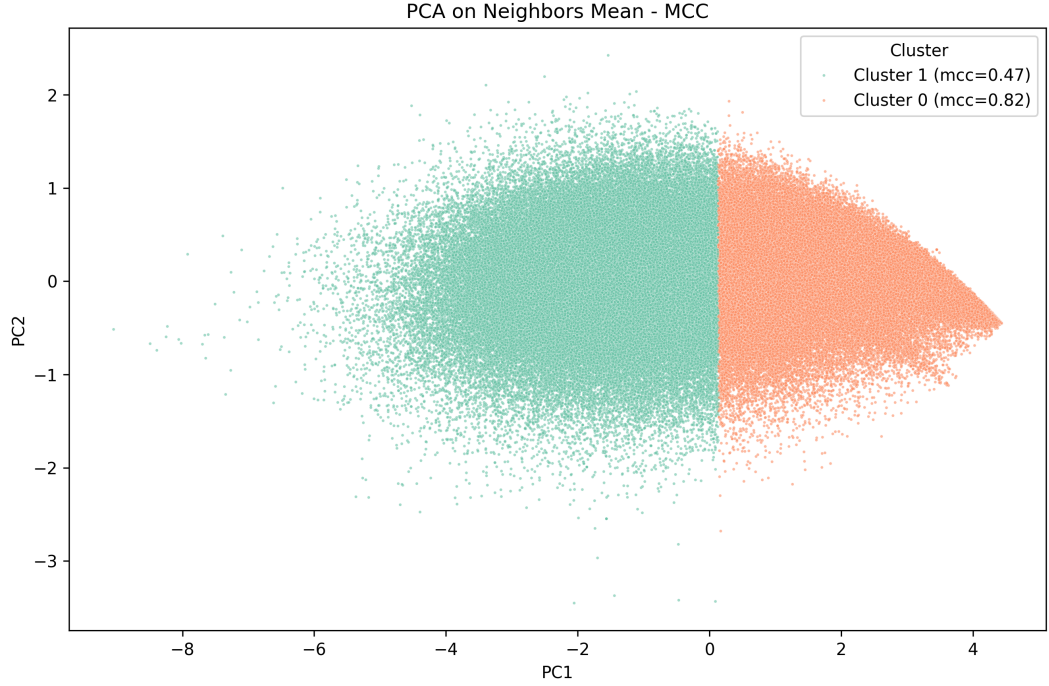


Figure 10: PCA on mean matrix using SpotSweeper-like analysis. For each classes MCC is measured and showed into the legend.

As observed in the PCA projection, the patches extracted from the nine patients are divided into two distinct groups, confirming that some patches are more informative than others for classification purposes. The most significant finding emerges from a comparison of performance across the two clusters: when evaluating the MCC—the metric chosen as the primary benchmark because, unlike accuracy, it accounts for the marked class imbalance present in the original dataset—cluster 1 achieves a value of **0.47** (with an accuracy of **65%**), while cluster 0 achieves an MCC of **0.82** (with an accuracy of **91%**). The clear performance gap between the two groups therefore confirms that the patches belonging to cluster 0 are significantly more informative for classification purposes than those in cluster 1. This indicates that these samples contain the key biological structures and morphological details necessary to effectively separate the clinical classes, standing out clearly from the more generic or non-specific cellular regions relegated to cluster 1.

Interestingly, the MCC of **0.82** achieved by cluster 0 represents a marked improvement over the baseline MCC of **0.60** achieved by the linear classifier on the entire set of patches, regardless of their informativeness. Indeed, when all

patches are considered without any form of discrimination, the model’s performance is significantly lower than that achieved by isolating only the patches identified as most informative. This confirms the hypothesis that much of the loss in overall performance is attributable to the presence, within the dataset, of a substantial number of patches of little diagnostic relevance.

UMAP analysis on high-purity samples Based on the analysis carried out using *SpotSweeper-like Analysis*, the vectors in the dataset were clustered into two categories of points: *informative* and *non-informative*.

Subsequently, a cross-validation of the test set’s quality was conducted to measure its spatial consistency in the latent representation space. Specifically, for each embedding of the held-out sample (excluding the sample itself from the count), its $k = 35$ nearest neighbors were identified among the 9 patients. For each point, the percentage of neighbors belonging to the cluster characterized by the highest performance values (cluster 0) was then calculated. A point in the test set was defined as ”pure” if this percentage exceeded the 70% threshold.

Finally, the points in the test set with purity above the threshold and those in cluster 0 of the 9 patients were plotted using the first two UMAP coordinates, calculated on the first 100 PCs, with the points colored according to the 3 macro-classes used by the classifier. This filtering allows the removal of background noise associated with uninformative or ambiguous patches, isolating only those samples with a highly coherent local neighborhood. This is expected to highlight greater spatial separation between classes in the UMAP latent space, favoring the emergence of compact and monochromatic clusters consisting predominantly of patches belonging to the same biological category.

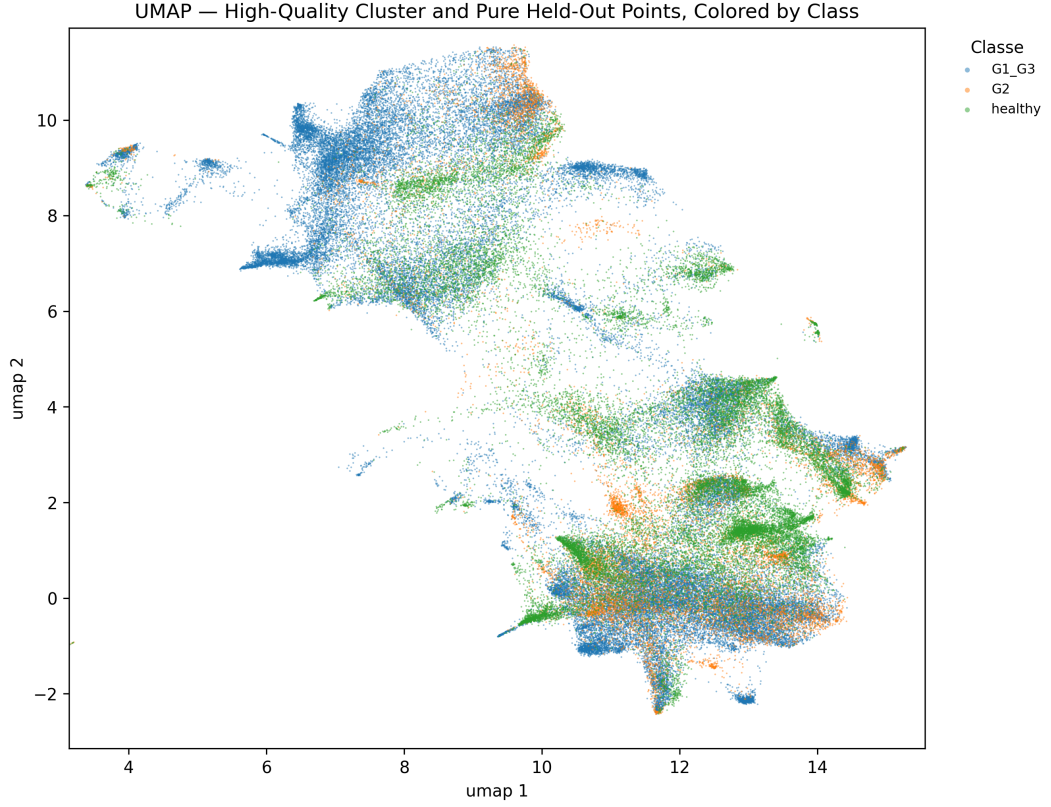


Figure 11: UMAP visualization of the *pure points* of the *test set* and the patches from cluster 0 obtained using *SpotSweeper-like Analysis*.

The UMAP visualization (Fig. 11) clearly shows that the model is able to capture the distinctive properties of the three classes, resulting in clusters composed almost exclusively of samples belonging to the same category.

The presence of homogeneous and well-separated clusters supports the effectiveness of self-supervised training, demonstrating that the architecture has learnt latent representations and ultrastructural features of great biological relevance without relying on labels during the training phase.

This finding provides a methodological explanation for the results obtained across the entire dataset: the initially modest classification performance does not stem from an intrinsic limitation of the model, but rather from the high incidence of noisy or indeterminate patches in the original dataset. Filtering out the noise using *SpotSweeper-like Analysis* and the purity criterion thus allows a marked spatial separability between the classes in the latent space to emerge.

5 Conclusions

This thesis explored two complementary methodological approaches for the automated analysis of TEM images, applied respectively to the tasks of mitochondrial segmentation and the morphological characterization of Gaucher disease: a supervised approach based on the U-Net architecture and a self-supervised approach based on SimSiam. For mitochondrial segmentation, the U-Net architecture achieved a class IoU of approximately **64.4%** despite the limited number of annotated images available—a result confirming the approach’s validity even under data—scarce conditions. The primary technical limitation encountered lies not in the architecture itself, but rather in the limited size and variability of the training dataset: the images used originated from a restricted set of acquisition conditions and exhibited consistently high visual quality. Expanding the dataset—both in terms of volume and diversity (by including different resolutions, instruments, and tissue types, as well as suboptimal or noisy samples)—would be the key step toward developing a more robust model capable of generalizing to real-world clinical scenarios.

As for the self-supervised learning approach, it yielded particularly encouraging results: without ever utilizing labels, SimSiam successfully captured the morphological information within the TEM images, despite the high complexity and level of detail characteristic of this data type. This is clearly evident in two respects. First, within the embedding space, the model arranges patches containing similar biological structures in adjacent regions, while separating patches with distinct structures into separate clusters. Second, a preliminary analysis revealed the model’s ability to generate monochromatic well separated clusters associated with individual patients; this suggests that the encoder learned to recognize features specific to different clinical classes, despite never having explicitly observed them during training.

This capability was also quantitatively confirmed by the linear classifier, which achieved an MCC of **0.60** on the entire validation dataset. When the evaluation is restricted to informative patches identified via SpotSweeper-like analysis, this figure rises to **0.82**, confirming that much of the accuracy loss observed on

the full dataset stems from the presence of non-discriminative or noisy samples, rather than an inherent representational limitation of the network. This suggests that for models operating on such rich and complex data —where useful and non-discriminative information coexist within the same image— it is highly advantageous to complement training with a strategy capable of filtering or weighting individual patches based on their actual relevance; *Multiple Instance Learning* (MIL) [28] represents one such approach and could serve as a natural extension for future work.

However, it should be noted that this performance is not maintained when the model is evaluated on a patient entirely excluded from the training set; in this scenario, the achieved accuracy is significantly lower. This drop in performance is likely attributable, at least in part, to the nature of the pathology itself —which lacks clearly distinctive morphological features for each severity class— and to class imbalance in the training dataset, which tends to systematically penalize the minority clinical class. Despite this limitation, the proposed approach demonstrates a solid foundation and promising results, serving as a valuable starting point for future analyses aimed at further improving the model’s generalization capabilities.

Bibliography

- [1] Yujian Mo et al. “Review the state-of-the-art technologies of semantic segmentation based on deep learning”. In: *Neurocomputing* (2022). DOI: 10.1016/j.neucom.2022.01.005.
- [2] Pendin D, Filadi R, and Pizzo P. “The Concerted Action of Mitochondrial Dynamics and Positioning: New Characters in Cancer Onset and Progression.” In: *Frontiers in Oncology* (2017). DOI: 10.3389/fonc.2017.00102.
- [3] Moon HE and Paek SH. “Mitochondrial Dysfunction in Parkinson’s Disease.” In: *Experimental Neurobiology* (2015). DOI: 10.5607/en.2015.24.2.103.
- [4] Itoh K et al. “Mitochondrial dynamics in neurodegeneration.” In: *Trends Cell Biol* (2012). DOI: 10.1016/j.tcb.2012.10.006.
- [5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation.” In: *arXiv* (2014). DOI: 10.48550/arXiv.1411.4038.
- [6] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. *ParseNet: Looking Wider to See Better*. 2015. arXiv: 1506.04579 [cs.CV]. URL: <https://arxiv.org/abs/1506.04579>.
- [7] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. *Learning Deconvolution Network for Semantic Segmentation*. 2015. arXiv: 1505.04366 [cs.CV]. URL: <https://arxiv.org/abs/1505.04366>.
- [8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: <https://arxiv.org/abs/1505.04597>.
- [9] Jie Gui et al. *A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends*. 2024. arXiv: 2301.05712 [cs.LG]. URL: <https://arxiv.org/abs/2301.05712>.

- [10] Stone W. L., Mukkamalla S. K. R., and S. R. Master. *Gaucher Disease*. In: *StatPearls [Internet]*. Treasure Island (FL): StatPearls Publishing. 2023. URL: <https://www.ncbi.nlm.nih.gov/books/NBK448080/>.
- [11] Bougea A. “Underlying Mechanisms of GBA1 in Parkinson’s Disease and Dementia with Lewy Bodies: Narrative Review.” In: *Genes* (2025). DOI: 10.3390/genes16121496.
- [12] E. Sidransky et al. “Multicenter Analysis of Glucocerebrosidase Mutations in Parkinson’s Disease”. In: *New England Journal of Medicine* (2009). DOI: 10.1056/NEJMoa0901281.
- [13] Hertz Ellen, Chen Yu, and Sidransky Ellen. “Gaucher disease provides a unique window into Parkinson disease pathogenesis”. In: *Nature Reviews Neurology* (2024). DOI: 10.1038/s41582-024-00999-z.
- [14] G. Imbalzano et al. “Neurological symptoms in adults with Gaucher disease: a systematic review”. In: *Journal of Neurology* (2024). DOI: 10.1007/s00415-024-12439-5.
- [15] National Gaucher Foundation. *Testing for Gaucher Disease*. URL: <https://www.gaucherdisease.org/gaucher-diagnosis-treatment/testing/> (visited on 08/06/2026).
- [16] Anna Dawid and Yann LeCun. “Introduction to latent variable energy-based models: a path toward autonomous machine intelligence”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2024.10 (Oct. 2024), p. 104011. ISSN: 1742-5468. DOI: 10.1088/1742-5468/ad292b. URL: <http://dx.doi.org/10.1088/1742-5468/ad292b>.
- [17] Kaiming He et al. *Masked Autoencoders Are Scalable Vision Learners*. 2021. arXiv: 2111.06377 [cs.CV]. URL: <https://arxiv.org/abs/2111.06377>.
- [18] Jane Bromley et al. “Signature Verification Using a ”Siamese” Time Delay Neural Network”. In: *International Journal of Pattern Recognition and Artificial Intelligence* (1993).

- [19] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG]. URL: <https://arxiv.org/abs/2002.05709>.
- [20] Xinlei Chen and Kaiming He. *Exploring Simple Siamese Representation Learning*. 2020. arXiv: 2011.10566 [cs.CV]. URL: <https://arxiv.org/abs/2011.10566>.
- [21] Mahmoud Assran et al. *Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture*. 2023. arXiv: 2301.08243 [cs.CV]. URL: <https://arxiv.org/abs/2301.08243>.
- [22] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: <https://arxiv.org/abs/1512.03385>.
- [23] PaddlePaddle Contributors. *PaddleSeg, End-to-end image segmentation kit based on PaddlePaddle*. 2019. URL: https://github.com/PaddlePaddle/PaddleSeg/blob/release/2.10/README_EN.md#-tutorials.
- [24] PyTorch. *resnet50*. URL: <https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html> (visited on 08/06/2026).
- [25] Totty M., S.C. Hicks, and B. Guo. “SpotSweeper: spatially aware quality control for spatial transcriptomics.” In: *Nature Methods* (2025). DOI: 10.1038/s41592-025-02713-3.
- [26] Williams CG et al. “An introduction to spatial transcriptomics for biomedical research.” In: *Genome Med* (2022). DOI: 10.1186/s13073-022-01075-1.
- [27] Matthijs Douze et al. “The Faiss library”. In: (2024). arXiv: 2401.08281 [cs.LG].
- [28] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. *Attention-based Deep Multiple Instance Learning*. 2018. arXiv: 1802.04712 [cs.LG]. URL: <https://arxiv.org/abs/1802.04712>.