Trabalho de Conclusão de Curso

Evaluation of Deep Metric Learning Methods for the Diagnosis of Human Visceral Leishmaniasis

Yanka Raíssa Ribeiro da Silva
yrrs@ic.ufal.br

Orientadora:

Profa. Dra. Fabiane da Silva Queiroz

Maceió, Dezembro de 2023

Yanka Raíssa Ribeiro da Silva

Evaluation of Deep Metric Learning Methods for the Diagnosis of Human Visceral Leishmaniasis

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientadora:

Profa. Dra. Fabiane da Silva Queiroz

Maceió, Dezembro de 2023

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

S586e Silva, Yanka Raíssa Ribeiro da.
Evaluation of deep metric learning methods for the diagnosis of human visceral leishmaniasis / Yanka Raíssa Ribeiro da Silva. – 2023.
53 f. : il.

Orientadora: Fabiane da Silva Queiroz.
Monografia (Trabalho de conclusão de curso em Ciência da Computação) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2023.

Bibliografia: f. 49-53.

1. Detecção de parasitas. 2. Leishmaniose visceral. 3. Aprendizagem profunda. 4. Redes neurais. 5. Classificação binária. 6. Diagnóstico automático de doenças. I. Título.

CDU: 004.032.26

Agradecimentos

Ao Primeiro Cientista e seus assistentes, pela orquestração dos fatos e eventos que culminaram nesse trabalho e os demais que derivarão a partir e por causa dele. Aqui eternizo a gratidão por toda honra e glória que me foi prometida.

À tríade sagrada, Irmã, Mãe, Avó, por me instigarem a curiosidade, a independência e a autenticidade. Minha fonte de tudo para além da vida.

A meu pai sagitariano.

À didática, paciência e prestatividade da *professora* Fabiane.

Aos amigos do VS, e também colegas de classe durante quatro anos. Se esse período foi fugaz e *quase* indolor foi porque me distraí e me apoiei na camaradagem e incontáveis memórias criadas juntos.

"Aceitai o que não conheceis, para que isso vos auxilie a conhecê-lo. Ressenti-vos contra ele, e continuará sendo um quebra-cabeça irritante."

– Mirdad

Resumo

A Leishmaniose Visceral, um tipo grave causado pelo complexo de parasitos *Leishmania donovani*, é fatal em mais de 95% dos casos não tratados e afeta predominantemente a população baixa renda, com acesso limitado a assistência médica. O exame parasitológico é o padrão ouro para o diagnóstico da LV; consiste na inspeção visual de amastigotas do parasita com cerca de 2-4 μm de diâmetro, o que pode rapidamente tornar-se uma tarefa exaustiva e exigir um nível de competência elevado. Visando auxiliar os profissionais de saúde, este estudo propõe uma abordagem alternativa que une aprendizagem métrica profunda a classificação supervisionada para a detecção rápida e precisa da leishmaniose visceral humana. A abordagem propõe dividir as imagens em conjunto de fragmentos (*patches*) para facilitar o discernimento durante a avaliação de quatro funções de perda de aprendizagem métrica profunda, visando a extração de características utilizadas por uma Máquina de Vetores de Suporte (SVM) para diagnosticar a leishmaniose visceral. Esse processo foi avaliado minuciosamente usando métricas relevantes como o Coeficiente de Correlação de Matthew (MCC), sensibilidade e especificidade, que revelaram que a função Circle supera o desempenho de outras funções com 98,3% de sensibilidade, 99,3% de especificidade e 97,7% de MCC. Em resumo, todas as funções avaliadas apresentaram um bom desempenho nas avaliações quantitativas, sugerindo que a aplicação da inteligência artificial no diagnóstico médico oferece benefícios consideráveis, especialmente ao auxiliar os médicos de forma economicamente eficiente na detecção rápida e precisa de doenças tropicais negligenciadas.

Palavras-chave: Deteção de Parasitas, Leishmaniose Visceral, Aprendizagem Métrica Profunda, Aprendizagem Profunda, Redes Neurais Convolucionais, Classificação Binária, Diagnóstico Automático de Doenças.

Abstract

Visceral Leishmaniasis, a severe type caused by the *Leishmania donovani* parasite complex, is fatal in over 95% of untreated cases and predominantly affects the poor and vulnerable with limited healthcare access. Parasitological processes are the gold standard for diagnosing VL; they entail direct microscopic inspection of amastigotes about 2–4 μm in diameter, which can quickly become a time-consuming, exhausting task and require an expert skill level. Aiming to assist physicians, this study proposes an alternative approach combining deep metric learning with supervised classification for the rapid and reliable detection of human visceral leishmaniasis. The suggested methodology segments images into patches for discernability during the evaluation of four deep metric learning loss functions to extract features, which are utilized by a Support Vector Machine (SVM) for the diagnosis of visceral leishmaniasis. This process was thoroughly assessed using key metrics like the Matthew Correlation Coefficient (MCC), sensitivity, and specificity, which revealed that Circle loss outperforms other losses with 98.3% sensitivity, 99.3% specificity, and 97.7% MCC. Overall, all of the functions evaluated performed well in quantitative assessments, implying that AI's application to medical diagnostics offers considerable benefits, particularly in cost-effectively assisting physicians in rapidly and accurately detecting neglected tropical diseases.

Key-words: Parasite Detection, Visceral Leishmaniasis, Deep Metric Learning, Deep Learning, Convolutional Neural Networks, Binary Classification, Automatic Disease Diagnosis.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

VL	Visceral Leishmaniasis
PCR	Polymerase chain reaction
qPCR	Quantitative real-time PCR
DML	Deep Metric Learning
PIL	Python Imaging Library
LMCL	Large Margin Cosine Loss
NSL	Normalized Softmax Loss
MS	Multi-Similarity Loss
CNN	Convolutional Neural Networks
SVM	Suport Vector Machine
PCA	Principal Component Analysis
MCC	Matthew Correlation Coefficient
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
ROI	Region of Interest
ReLU	Rectified Linear Unit
DOT	Dot Product Similarity
COS	Cosine Similarity

Conteúdo

1

Introduction

1.1 Human Visceral Leishmaniasis

Leishmaniasis is a neglected and contagious vector-borne disease caused by species of the intracellular protozoan genus *Leishmania*. It is prevalent in the poorest countries and among the most vulnerable individuals with little access to health treatment. Visceral Leishmaniasis (VL), a more severe Leishmaniasis also known as kala-azar, is caused by the *Leishmania donovani* complex, a group of parasite species that is the principal cause of this potentially fatal disease.

If not recognized and treated, VL is fatal in more than 95% of patients. The World Health Organization (WHO)[1] describes it as causing irregular bouts of fever, weight loss, spleen and liver enlargement, and anemia. In 2021, 99 nations and territories were known to be endemic to Leishmaniasis, with 81 countries endemic to VL. In the Americas, VL is endemic in 12 countries. South American countries, such as Brazil, Argentina, Colombia, Paraguay, and Venezuela, have among the highest case records.

Honduras and Guatemala, for instance, previously reported sporadic VL cases but indicated an increasing number of cases in 2022 (WHO TEAM and Services, 2023). In southern Europe, it is a primary opportunistic infection in patients with acquired immunodeficiency syndrome (Peters et al., 1990). The majority of cases are found in Brazil, East Africa, and India. As a result, more than 1 billion people live in Leishmaniasis-endemic areas and are at risk of infection. It is estimated that 50,000 to 90,000 new cases of VL are diagnosed each year worldwide, with only 25 to 45 percent being reported to the WHO.

Visceral Leishmaniasis is diagnosed using DNA-based and non-DNA-based methods (Akhoundi et al., 2017). DNA-based methods, like PCR and qPCR, are complex and expensive, limited to a few teaching hospitals and research facilities in VL-endemic countries (Antinori et al., 2007; Kumari et al., 2021). On the other hand, non-DNA-based approaches, such as

[1] https://www.who.int/health-topics/leishmaniasis

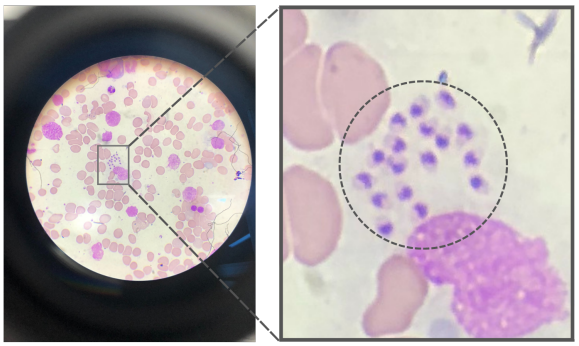


Figura 1: Example of an image captured from bone marrow smears. The zoomed circular area indicates the presence of *Leishmania* amastigotes.

serological methods and parasitological procedures, detect antibodies or antigens but may lack specificity for asymptomatic infections that require a specific serological diagnosis (Werneck et al., 2002; Kumari et al., 2021).

According to Erber et al. (2022), parasitological techniques are the gold standard for diagnosing VL. It comprises direct microscopic inspection of the parasite's amastigote form in aspired/biopsied tissues such as bone marrow, lymph nodes, and spleen (van Griensven and Diro, 2019). Smears are simple to conduct, and their direct examination is usually the best diagnostic method in more impoverished areas where PCR is not available (Elmahallawy et al., 2014). Figure 1 presents an example of *Leishmania* amastigotes in a bone marrow microscopic color image.

Reimão et al. (2020) describes *Leishmania* amastigotes as intracellular round or oval bodies, about 2–4 μm in diameter, with distinctive nuclei and kinetoplasts. As previously stated, parasitological processes entail direct microscopic inspection of these minuscule amastigotes, which can quickly become a time-consuming, exhausting task and require an expert skill level (Srivastava et al., 2011). That is because, faced with the existence of the protozoan, the physician may be unsure whether it is a *Leishmania* given that it may resemble other structures in the image content.

As a result, its sensitivity is quite poor. The more secure procedure is to obtain a biopsy from the bone marrow and examine the material stained with Giemsa[2]. Still, the sensitivity of this procedure is about 60% to 85% (Elmahallawy et al., 2014).

To alleviate repetitive work, machine learning techniques are being used to process medical images for disease diagnosis, offering advantages over traditional manual methods. These include faster analysis, reduced variability, automated processing of large data volumes, and the detection of subtle patterns. Computer Vision and Deep Learning are particularly useful in detecting diseases, including VL in humans, and achieving high precision in analyzing bone marrow microscopy images.

[2]Giemsa's staining solution is one of the most common microscopic stains, generally used in hematology, histology, cytology, and bacteriology for in vitro diagnostic.

1.2 Objectives

This study seeks to evaluate the impact and effectiveness of deep metric learning methods in accurately diagnosing human visceral leishmaniasis using microscopic images. This evaluation is underpinned by several specific objectives. First, the study conducts an experimental approach to accentuate areas of relevance within the images and segment these images into smaller patches, with the expectation that the significant features of the images will become more discernible.

Second, the study compares various deep metric learning algorithms to pinpoint the most effective models for extracting features. This comparison aims to determine which models are best suited for the nuanced task of identifying and interpreting complex patterns associated with the disease.

Third, the research configures a supervised classification algorithm to categorize images based on the data extracted from the metric learning models. It's expected that the trained classifier will become an expert in translating the characteristics extracted by the deep metric learning models into actionable diagnostic insights.

Finally, the study assesses key performance metrics, including the Matthew Correlation Coefficient, sensitivity, and specificity of the classifier. This evaluation is focused on determining the classifier's effectiveness in image classification for VL diagnosis, ensuring that it can reliably distinguish between positive and negative cases of the disease.

The findings should make an important contribution to the field of artificial intelligence applied in medical analysis, particularly in assisting physicians in detecting neglected tropical diseases more rapidly and reliably at an inexpensive cost.

1.3 Related Works

Most parasitic protozoans that affect humans are no more than 50 μm in size (Reimão et al., 2020). As a result of their significantly smaller size, they provide a considerable challenge to diagnosis via microscopy image evaluation (Srivastava et al., 2011). *Plasmodium*, *Trypanosome*, *Babesia*, *Toxoplasma*, *Leishmania*, and *Trichomonad* are well-known disease-causing protozoan parasites. In recent years, several ways to parasite examination from microscope images have been presented. Recent reviews of the published literature can be found in Liu et al. (2021); Zhang et al. (2022).

In general, microscopy image analysis includes object detection (Yang et al., 2020; Koirala et al., 2022), segmentation (Salazar et al., 2019; Yang et al., 2022), tracking (Spilger et al., 2021), and image reconstruction (Qin, 2022) methods. Classification methods include cell type differentiation and are typically used for object detection (Liu et al., 2021). This section will focus on object detection and segmentation of individual parasites in microscopy images.

1.3.1 Parasite Detection

Yang et al. (2020) and Fuhad et al. (2020) have proposed ways for detecting malaria parasites in thick blood smears using smartphones. Yang et al. (2020), describes a procedure in two steps: To select parasite candidates, researchers initially used an intensity-based Iterative Global Minimum Screening (IGMS), which provides a quick screening of a thick smear image. Following that, each candidate was classified as a parasite or background using a modified Convolutional Neural Network (CNN). Fuhad et al. (2020) developed a variety of accurate and computationally efficient models for parasite detection in single cells. The simplified variant was also used in mobile phones and a server-backed online application.

Soberanis-Mukul et al. (2013) achieved an automated approach for detecting *Trypanosoma cruzi* parasites in digital microscope pictures derived from peripheral blood smears stained with Wright's stain. Authors suggest combining image pre-processing algorithms such as binary mask generation, Gaussian filtering, and domain intersection with a KNN classifier applied across a segmented part of the original image.

1.3.2 *Leishmania* Detection

The gold standard for diagnosing VL in humans is images from bone marrow parasitological examinations, as recommended by the WHO (WHO TEAM, 2023). In the state of the art, a few works implement an automated parasite examination over images from bone marrow smears.

Farahi et al. (2015) use morphological and CV level set approaches to segment *Leishmania* bodies in digital color microscopic images recorded from bone marrow samples. Salazar et al. (2019) proposes a semiautomatic segmentation approach for obtaining the segmentation of the evolutionary forms of Visceral Leishmaniasis parasites. Smoothing filters and edge detectors improve the optical microscopy pictures, and segmentation is performed via a region-growing algorithm.

Isaza-Jaimes et al. (2021) propose a detection method that uses image processing techniques, like low-pass filters, gradient operators, and gradient modules based on polar maps of the pixel intensities. Coelho et al. (2020) uses morphological mathematical operators to segment the parasites.

Górriz et al. (2018) present a non-supervised model-based method for segmentation of *leishmania* parasites in microscopy images from bone marrow smears. For that, they trained a U-net model Ronneberger et al. (2015) (Deep Learning-based approach) that successfully segments parasites and classifies them into promastigotes, amastigotes, and adhered parasites.

Along the same lines, Gonçalves et al. (2023) employed a U-Net architecture to automatically pinpoint the pixels of interest in the images, in this context, those containing *Leishmania* parasites. This process was guided by binary masks annotated by specialists.

The experiments of Farahi et al. (2015); Salazar et al. (2019); Isaza-Jaimes et al. (2021) were performed over a public dataset provided by Farahi et al. (2015) whereas Ronneberger

et al. (2015); Górriz et al. (2018); Gonçalves et al. (2023) conducted their experiments in non-public datasets.

1.4 Work Structure

The overall structure of this study takes the form of five chapters. Chapter 2 begins by laying out the theoretical dimensions of the research, highlighting the use of image processing and computer vision in the study of medical images.

Chapter 3 is concerned with the methodology used for this study.

The emphasis in Chapter 4 changes to data collecting and processing, culminating in a comprehensive discussion of the research findings.

Chapter 5, which terminates the study, summarizes the general accomplishments of this review, relating them to the broad and specific objectives established at the outset. It also provides insights into prospective study directions.

2

Theoretical Background

2.1 Deep Learning

Deep Learning is the subfield of Artificial Intelligence that attempts to simulate the behavior of the human brain by focusing on extracting features in data, especially unstructured data such as images and text. Those models are capable of making accurate data-driven decisions and are particularly suited to contexts where the data is complex and there are large datasets available (Kelleher, 2019). Significantly, in the healthcare sector, Deep Learning has shown immense potential, especially in processing medical images (X-rays, CT, and MRI scans) to diagnose health conditions (Bakator and Radosav, 2018).

The representation of the data presented to Artificial Intelligence systems has a significant impact on their performance (Dodge and Karam, 2016). A feature is any component of information that is included in the representation of data instances, therefore, many AI tasks can be handled by first determining the best set of features to extract for that task, and then feeding those characteristics into a simple Machine Learning algorithm. Specific keywords or sender reputation, for example, can be critical in Spam Email Identification (Yaseen et al., 2021).

However, it is difficult to discern which features should be taken from high-level data. Image Identification is a good illustration of this, as determining essential features for discriminating between thousands of item categories can be quite challenging (Pak and Kim, 2017). In that context, Deep Learning enables the computer to build complex concepts out of simpler concepts by breaking the desired complicated mapping into a series of nested simple mappings, each described by a different layer of the model (Goodfellow et al., 2016).

The input is displayed in the visible layer, so-called because it contains the variables that one can see. The image is then extracted into a series of hidden layers, which capture progressively abstract information. These layers are referred to as "hidden"because values are not provided in the data; instead, the model must infer which ideas are relevant for understanding

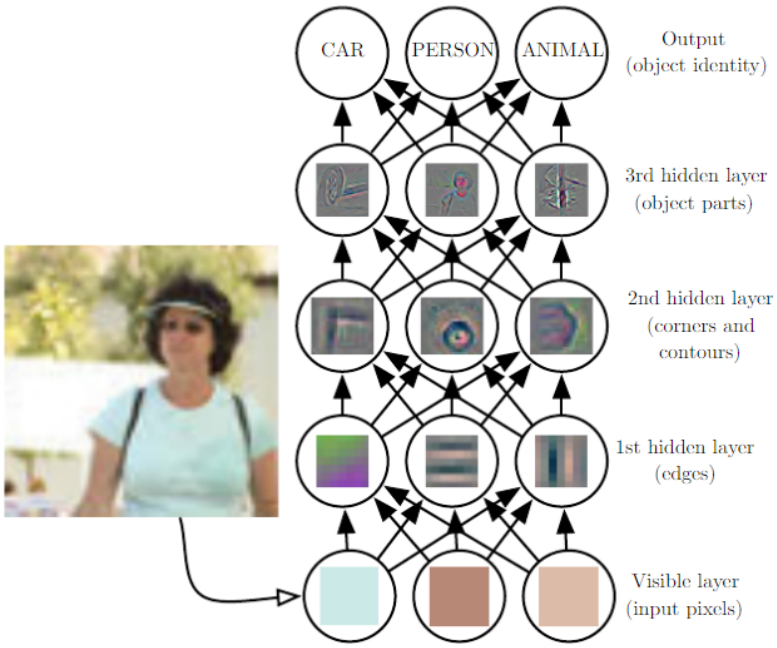


Figura 2: Illustration of a Deep Learning model. The images here depict the type of feature represented by each hidden unit. The first layer can readily identify edges given the pixels by comparing the brightness of neighboring pixels. Given the description of the edges, the second hidden layer may easily search for corners and extended contours, which are recognized as collections of edges, and so on for the subsequent layers. The description of the image in terms of its object parts can be utilized to identify the items present at the end of the last layer. Source: Goodfellow et al. (2016).

the observed data's relationships. Figure 2 illustrates a typical Deep Learning model applied to Object Detection.

2.1.1 Image Processing and Computer Vision

At the core of Image Processing is the manipulation of digital images through various algorithms to enhance image quality or to extract useful information (Ritter et al., 2011). This process frequently includes operations such as filtering, image enhancement, noise reduction, and image restoration. Computer Vision, on the other hand, goes beyond simple image processing to enable machines to comprehend and make judgments based on visual data.

The goal is to enable machines to recognize patterns, identify objects, and comprehend situations in images and videos to recreate the complexity of human vision. With the introduction of Deep Learning, both fields have advanced substantially. Modern algorithms, particularly those based on neural networks, have transformed the way images are processed and interpreted.

Convolutional Neural Networks (CNNs), which have emerged as a cornerstone architecture, are one of the most commonly used architectures in Computer Vision. CNNs, inspired by the organization of the human visual cortex, are designed to learn spatial hierarchies of fe-

atures from input images automatically and adaptively. A typical CNN architecture consists of several layers, including convolutional layers that apply filters to the input for feature extraction, pooling layers that reduce data dimensionality, and fully connected layers that perform classification based on the extracted features.

The core building block of a CNN, the convolutional layer, employs various filters to capture different aspects of an image, such as edges or textures. Each filter generates a feature map that highlights these aspects. Following pooling layers reduce the spatial size of these feature maps, reducing computational load and achieving translational invariance. Lastly, these feature maps are interpreted by the fully connected layers to make predictions or classifications (Goodfellow et al., 2016).

CNNs have become pivotal in Computer Vision tasks such as Image Classification (Sundgaard et al., 2021), Object Tracking (Hu et al., 2015) , and Face Recognition (Schroff et al., 2015).

2.1.2 Medical Images Analysis

Another significant aspect of Image Processing and Computer Vision algorithms is their broad application in Medical Image Analysis. The combination of advanced computational techniques and Medical Imaging has resulted in more precise and efficient analysis, which has greatly aided in Disease Detection, Diagnosis, and Treatment Planning.

Image Processing in Medical Imaging encompasses a variety of techniques, all of which aim to improve the interpretability of the depicted contents (Ritter et al., 2011). Image Enhancement is used to improve visual quality, Segmentation is used to isolate specific regions or structures (Liu et al., 2022), such as an organ, and Feature Extraction is used to identify unique attributes within images. This processing ensures that the data is effectively interpreted by the subsequent Computer Vision methods .

One of the most significant applications of CNNs in Microscopic Image Analysis is in the field of Hematology, where they are used in Cell Type Classification, Stem Cell Motion Tracking, and Diagnosis of Blood-Related Diseases (Liu et al., 2021).

2.1.3 Deep Metric Learning

Metric Learning is an approach based directly on a distance metric that aims to reduce the distance between similar objects and simultaneously increase the distance between dissimilar objects (Kaya and Bilge, 2019). The method is based on a W projection matrix where the data is moved to the transformation space with distance information. Thus, Deep Learning and Metric Learning have been combined in recent years to establish the concept of Deep Metric Learning (DML) as depicted in Figure 3.

DML is to explicitly learn a nonlinear mapping f to map data points into a new feature space

by leveraging deep neural network architecture, in which f is parameterized by deep neural network weights and biases (Lu et al., 2017). Hence, the integration of metric loss functions, sampling methodologies, and network topology is at the heart of Deep Metric Learning. This holistic approach to network design and operation considers the relationships between samples as dictated by the metric loss function.

There are many different suggestions for loss functions, such as contrastive loss (Hadsell et al., 2006), triplet loss (Schroff et al., 2015), quadruple loss (Ni et al., 2017), and n-pair loss (Sohn, 2016). The definition of an appropriate loss function ensures fast convergence and optimizes the global minimum search. The aforementioned functions enable the expansion of data sample sizes in forms like paired samples (n^2), triplet samples (n^3), and quadruple samples (n^4).

They share the basic premise of optimizing the distances between pairs or groups of examples inside the learned feature space. They train the neural network to incorporate the data in a domain where similarities and dissimilarities may be quantified using metrics like Euclidean distances or cosine similarities (Lu et al., 2017). As a result, crucial patterns and structures arise in the data representation, allowing the network to find nuances and connections that would be difficult to distinguish in less precise feature spaces. As a result, these loss functions enable the network to capture and reflect the data's complexity and richness, making them valuable tools for jobs requiring fine discrimination of similarities and differences.

Deep Metric Learning has a wide range of applications, including Person Reidentification (Yi et al., 2014), Chest Radiograph Analysis (Zhong et al., 2021), and Object Tracking (Hu et al., 2015). In the context of medical diagnostics, the potential of DML is vast, offering promising avenues for research and development. This is exemplified by innovative applications such as Brain Tumor Segmentation (Liu et al., 2022) and Tympanic Infection Detection (Sundgaard et al., 2021).

The sections that follow distinguish four examples of DML loss functions that are essential for this study.

2.1.4 Triplet

Schroff et al. (2015) demonstrated that the triplet loss function is a framework designed to understand and quantify the relationship between three principal data points: an anchor, a positive, and a negative. The anchor serves as the reference point, the positive is another data point that shares similarities with the anchor, and the negative is distinct from the anchor. The driving goal behind this methodology is to minimize the feature space distance between the anchor and the positive while maximizing the distance between the anchor and the negative. This is achieved by ensuring the former is less than the latter by a predefined margin.

Mathematically, the triplet loss is defined as the maximum of the difference in distances plus a margin and zero, which can be represented as:

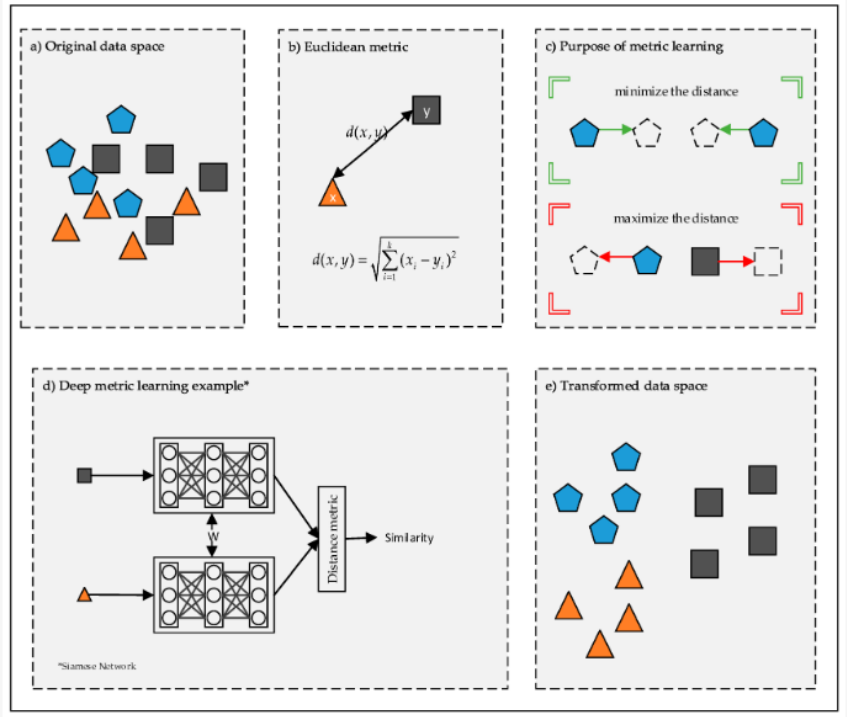


Figura 3: Deep Metric Learning. Source: Kaya and Bilge (2019).

$$\sum_{i=1}^{N} \left[\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 + \alpha \right]_+$$

In this expression, the output denotes the sum over all triplets N in the training set, where x_i^a, x_i^p, and x_i^n represent the anchor, positive, and negative of a triplet, respectively. The function $f(x)$ represents the embedding of the image into Euclidean space, α is the margin between positive and negative classes, and $[\cdot]_+$ indicates the hinge loss function, which is zero for negative arguments, enforcing the condition that the loss is non-negative.

The triplet loss function needs a strategic training approach in which the model's parameters are continually adjusted to reduce the loss. This optimization process inherently teaches the model to align positive examples closely with the anchor and to alienate the negative examples, thus refining the model's predictive accuracy.

2.1.5 Circle Loss

While the Triplet loss has been instrumental in enhancing the model's predictive accuracy by optimizing distances within the embedding space, it is not without its limitations. Specifically, its rigidity in gradient allocation and the potential for ambiguous convergence points suggest the necessity for a more flexible optimization strategy. The Circle Loss addresses these concerns by introducing a more adaptable gradient system that differentiates between similarity scores based on their proximity to the optimum.

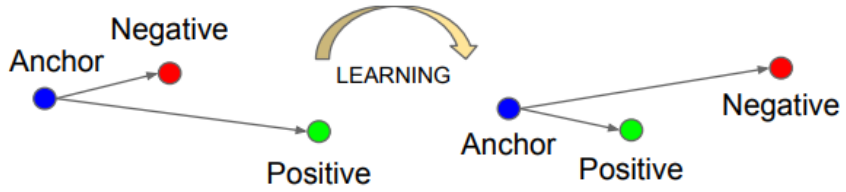


Figura 4: The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity.

It is designed to maximize within-class similarity and minimize between-class similarity, addressing the limitations of traditional loss functions like softmax cross-entropy (Sun et al., 2020). These conventional loss functions embed similarities into pairs and aim to reduce the difference between them, which leads to inflexible optimization due to equal penalty strength across all similarity scores. Circle Loss re-weights each similarity score to emphasize less-optimized similarities, resulting in a more flexible optimization process with a circular decision boundary.

The Circle Loss function provides a unified formula for two fundamental deep feature learning paradigms: learning with class-level labels and learning with pair-wise labels. Through dynamic adjustment of gradients during training, the less-optimized similarity scores receive larger weighting factors, leading to larger gradients and more effective updates.

The decision boundary in Circle Loss is circular in the similarity pair space, which simplifies to a point on the boundary for convergence, setting a definite target. This is a departure from the ambiguous convergence status of other loss functions, where any point along a linear decision boundary is acceptable. Mathematically, Circle Loss is expressed as:

$$L_{circle} = \log\left(1 + \sum_{i=1}^{K}\sum_{j=1}^{L} \exp\left(\gamma(\alpha_{jn}s_{jn} - \alpha_{ip}s_{ip})\right)\right)$$

where α_{jn} and α_{ip} are non-negative weighting factors, s_{jn} and s_{ip} are the between-class and within-class similarity scores, K and L are the size number of positive and negative class sample set, and γ is a scale factor that controls the strength of penalization.

2.1.6 Multi-Similarity

Another significant loss function proposed within the General Pair Weighting (GPW) was introduced by Wang et al. (2019).

The Multi-Similarity loss specifically addresses the challenge of sampling informative pairs for training, which is crucial for the success of pair-based deep metric learning methods. It does so by considering three types of similarities: self-similarity, positive relative similarity, and negative relative similarity. These similarities measure the relevance of the pairs and are

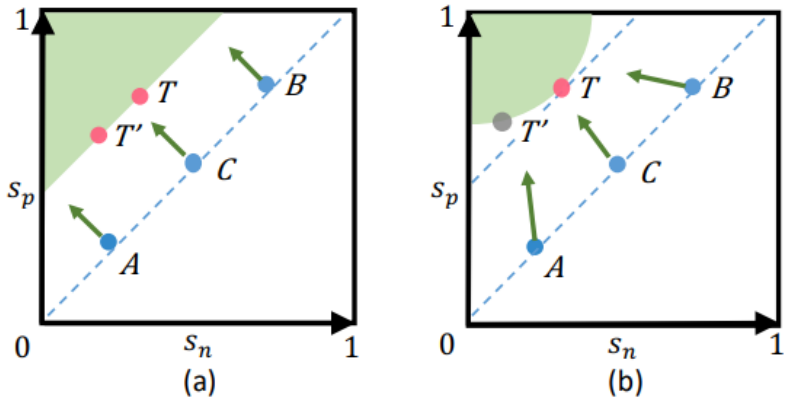


Figura 5: Comparison between the popular optimization manner of reducing $(s_n - s_p)$ and the proposed optimization manner of reducing $(\alpha_n s_n - \alpha_p s_p)$. (a) Emphasizes on increasing s_p. (b) Emphasizes reducing s_n Moreover, it favors a specified point T on the circular decision boundary for convergence, setting up a definite convergence target.

used to weigh them during the learning process.

Self-similarity is the intrinsic similarity within a pair, positive relative similarity is the similarity of a pair compared to other positive pairs, and negative relative similarity is compared to other negative pairs. The MS loss aims to maximize the self-similarity for positive pairs and minimize it for negative pairs while also considering the relative similarities to ensure that the pairs are optimally weighted.

Mathematically, the MS loss function is defined using an iterative process of mining and weighting. Informative pairs are first sampled using a mining strategy based on positive relative similarity, and then these pairs are weighted more precisely by considering both self-similarity and negative relative similarity. The MS loss is formulated as follows:

$$L_{MS} = \frac{1}{m}\sum_{i=1}^{m}\left(\log\left(1+\sum_{k\in P_i} e^{-\alpha(S_{ik}-\lambda)}\right)^{\alpha} + \log\left(1+\sum_{k\in N_i} e^{\beta(S_{ik}-\lambda)}\right)^{\beta}\right)$$

Here, α and β are hyper-parameters controlling the strength of the weight for positive and negative pairs, respectively, and λ is a margin parameter. S_{ik} represents the cosine similarity between the embedding of the anchor sample i and a sample k, and P_i and N_i are the sets of positive and negative pairs related to the anchor i.

2.1.7 NPairs

Finally, Sohn (2016) proposed the N-pair loss function that extends the classic triplet loss by comparing a positive example against multiple negative examples simultaneously. The multi-class N-pair loss function, denoted as N-pair-mc loss, is designed to optimize the identification of a positive example from multiple negative examples. It addresses the key limitation of triplet

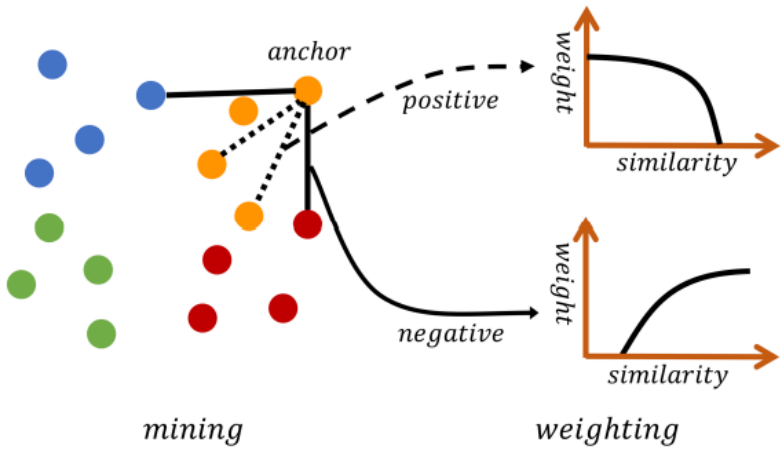


Figura 6: Objective of the Multi-similarity loss, which aims to collect informative pairs, and weigh these pairs through their own and relative similarities.

loss which only considers one negative example at a time, failing to account for the distribution of the remaining negative classes. By incorporating N-1 negative examples, this loss function ensures that the embedding for a given instance is distinct from multiple negative classes, promoting a more stable and balanced metric learning process.

Mathematically, the N-pair-mc loss is formulated as:

$$L_{\text{N-pair-mc}}((x_i, x_i^+)_{i=1}^N; f) = \frac{1}{N}\sum_{i=1}^{N} \log\left(1 + \sum_{j \neq i} \exp(f(x_i)f(x_j^+) - f(x_i)f(x_i^+))\right)$$

where x_i represents the anchor input feature vector for the i-th example in a batch, x_i^+ denotes the positive example that is similar to the anchor input x_i and belongs to the same class, x_j^+ refers to negative examples that are dissimilar to the anchor input x_i and belong to different classes. These are the features against which the anchor is compared within the loss function. Lastly, N indicates the number of distinct classes represented in a batch.

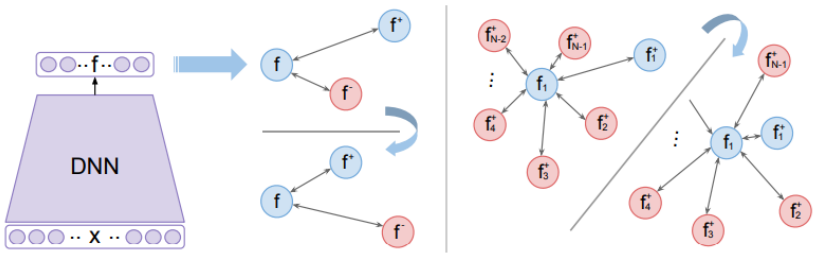


Figura 7: Triplet loss (left) pulls a positive example while pushing one negative example at a time. On the other hand, (N+1)-tuplet loss (right) pushes N-1 negative examples all at once, based on their similarity to the input example.

2.2 Classification Tasks

In the domain of machine learning, tasks are typically defined by the requirements of how the system should process a given example. One of the fundamental tasks in machine learning is classification. In this task, a computer program is tasked with determining which of k distinct categories an input belongs to.

To accomplish this, the learning algorithm typically develops a function f, which, when applied, assigns an input represented by vector x to a category indicated by the numeric code y. Variants of this task include those where f outputs a probability distribution over classes (Goodfellow et al., 2016).

In the context of supervised learning, algorithms are exposed to a dataset containing features, with each sample matched with a label or target. A significant example in the medical area is the classification of patients for diabetes risk based on several physiological data. In this case, a supervised learning algorithm can examine the dataset to distinguish between patients at high and low risk for diabetes based on their medical data (Butt et al., 2021).

To assess the capabilities of a machine learning algorithm, a quantitative performance measure must be developed. This skill is critical for understanding how well the algorithm will perform with real-world, previously unknown data. As a result, performance evaluation is carried out using a separate dataset from that used to train the machine learning system. In this review procedure, various measures such as precision, accuracy, and sensitivity are used.

Certain measures may have more weight in measuring performance in each given scenario. In medical diagnostics, for example, sensitivity (true positive rate) becomes an essential parameter, particularly for dangerous illnesses such as cancer or heart disease. The algorithm's high sensitivity ensures that it accurately detects the majority of individuals who have the disease, which is critical to avoid missing a diagnosis in potentially life-threatening situations.

2.2.1 Support Vector Machines

In that context, Support Vector Machine (SVM) is a supervised learning model that is commonly used in classification and regression tasks. Its major characteristic is the capacity to discover the hyperplane or group of hyperplanes in a high or infinite dimensions space that optimally separates the distinct classes of data.

With the application of kernel functions, data can be converted into a higher-dimensional space with linear separation. Then, SVM shifts the hyperplane to maximize the distance between support vectors, data points nearest to the hyperplane, and the hyperplane, improving class separation. This makes SVM suited for data sets with numerous variables since it is adaptable to diverse types of data, including ones with non-linear relationships. Figure 8 demonstrates the result of this process.

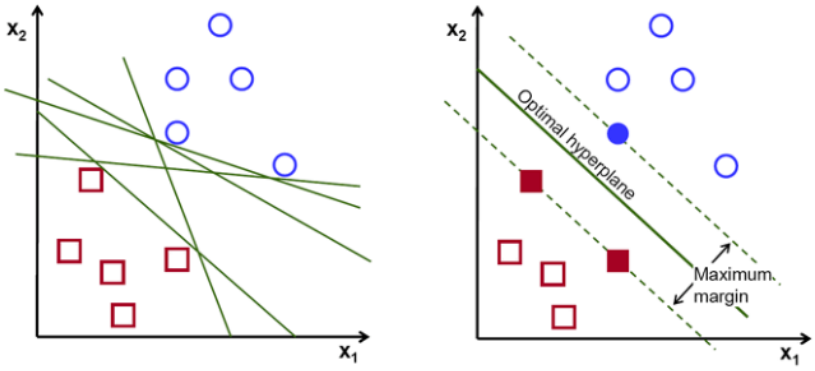


Figura 8: Illustration of a Support Vector Machine.

3

Methods

This chapter outlines a procedure that outputs a binary classification for the presence of amastigotes in images. Accurate diagnosis is paramount, as a high rate of false negatives (FN) can lead to insufficient or delayed treatment, consequently worsening the illness and impairing the patient's prognosis. Figure 9 illustrates the methodology of the feature extraction and classification process.

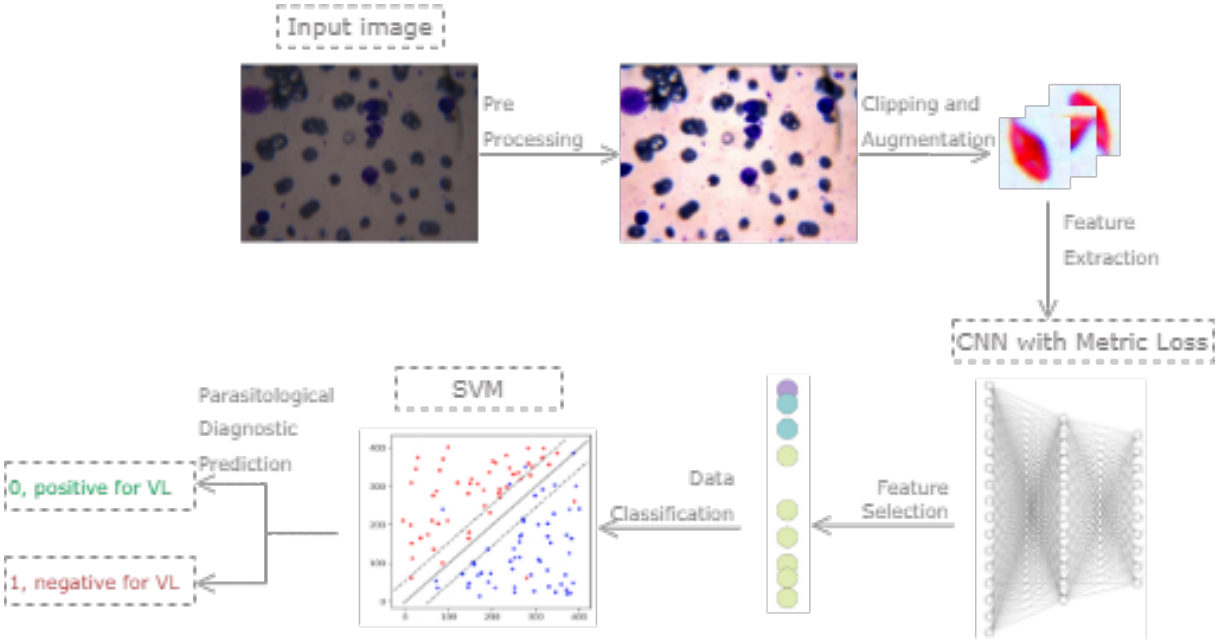


Figura 9: Flowchart depicting the process from image acquisition to parasitological diagnosis using CNN with metric loss and SVM classification for VL detection.

3.1 Preprocessing

Pre-processing refers to the initial steps of image treatment before they are analyzed by the model, and it was performed on biological sample full images in the presence of amastigotes.

Contrast is essential in micrography for identifying and quantifying individual structures. These tasks can be imprecise or perhaps impossible to complete without appropriate contrast. As a result, contrast enhancement is a digital image processing tool that modifies the intensity values of pixels. This can be accomplished by raising the intensity difference between the image's lightest and darkest pixels, resulting in a more visually clear image.

The method used in this research, linear interpolation, enables the original image to be blended with a modified version of itself in which pixel intensity values are altered to improve contrast. The linear interpolation formula is used to do the following:

$$out\_img = original\_img \times (1 - \alpha) + altered\_img \times \alpha$$

Where α is the contrast factor dictating the degree of enhancement. The increased contrast image thus shows more distinct cellular features and a greater dynamic range of intensities, making image analysis easier. Therefore, α was assigned a value of 1.5.

3.2 Dynamic Image Clipping

As stated in Section 1.1, the *Leishmania* parasite represents a small dot on the image, with a proportion of 3% to 5% of the image size. Thus, using images with real dimensions in the metric learning model implies the problem of losing information about the amastigotes' pixels due to the reduction of the dimensionality of the images to be entered into the network. Thus, a dynamic stride image clipping was performed on the images to avoid this problem. The graphical overview of the clipping algorithm is described in Figure 10.

Building upon the methodology reported by Gonçalves et al. (2023), which segments images and binary masks into smaller clippings by dynamically adjusting the step based on the presence of target features, this study adopts a similar strategy. In this approach, 96x96 pixel clippings are generated by traversing binary masks that contain annotations indicating the locations of amastigotes in RGB images. When a marked area is encountered in the mask, the step of the window is decreased to an eighth of its original size. Subsequently, the area of the *Leishmania* within this region is analyzed. If this area exceeds a predetermined threshold, the resulting clipping is categorized as belonging to the positive class. Conversely, if the area falls below this threshold, it is classified as negative.

Every parameter related to the clipping process, including the size of the clippings, the interval between them, and the smallest area required for labelling as *Leishmania* positive, has been established in advance. Furthermore, a series of evaluations was conducted to identify the

most suitable parameters for the dataset used in this research. The outcomes of these evaluations are concisely presented in Table 1.

Hyper-parameter	**Value**
Dimensions of the clippings	96x96
Step between clippings with the presence of amastigotes	12 pixels
Step between clippings with absence of amastigotes	96 pixels
Minimum area of *Leishmania* inside the clipping (α)	20%

Tabela 1: Hyper-parameters tested and used for cropping the images.

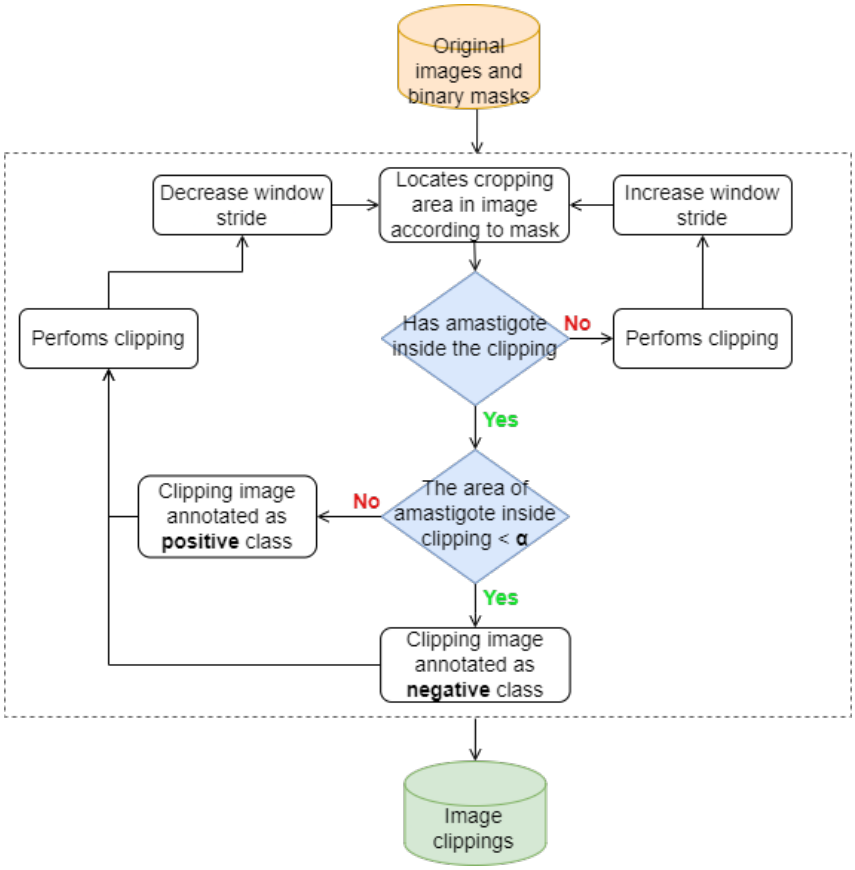


Figura 10: A schematic of the method used for slicing images, with the dotted line area illustrating the repeated cycles of the clipping algorithm.

Although effective, the clipping algorithm does not resolve the challenge of the limited presence of leishmania parasites in the images by itself, resulting in a skewed production of fewer positive class clippings compared to the negative class. To address this imbalance, the subsequent section will delve into data augmentation techniques to artificially enhance the dataset.

3.3 Data Augmentation and Class Balancing

As outlined in existing literature, data augmentation is a technique in machine learning used to generate additional training samples by applying geometric transformations to existing data. It aims to enhance model robustness and performance, especially when dealing with small or non-representative datasets.

To avoid undesirable image distortions, a strategy to follow is to identify which techniques best fit the situation and which hyper-parameter ranges are allowed. Thus, for each image belonging to the positive class, synthetic synthetic data were produced based on a set of specified transformations, such as rotating each image by up to 120 degrees, applying both horizontal and vertical flips, and zooming up to 10%.

However, augmented data is, essentially, not new information but merely reiterations or variations of minority class data. Relying too heavily on such augmented data can cause the model to learn features that are specific to the synthetic samples rather than generalizing from the actual distribution of data. Therefore, given the potential pitfall of overfitting through excessive synthetic data to match the proportion of positive-negative classes, an additional step in this process is to downsample the majority class (the negative class).

Downsampling involves randomly removing k samples from the majority class. In the context of this study, k was set to be a number that $k = N - 2P$, with N and P being the size of the negative and positive classes, respectively. Hence, the final proportion of the dataset will be 1:2.

3.4 Feature Extraction

Having discussed the image-related processing of the dataset, this section delves into the methodology employed for extracting feature representations using CNN and DML. The efficacy of a CNN in such tasks is significantly influenced by the choice of the loss function during training, which guides the network toward learning discriminative features that are crucial for the task at hand.

After analyzing the theoretical background of this work, Triplet, Circle, MultiSimilarity, and NPairs were chosen since they appeared frequently among the choices of authors of related works. Except for Circle loss, which is an independent choice, incorporated after reviewing the mathematical basis. Each of them was trained with the same CNN architecture and compared against each other. Every method is intended to maximize the feature space distinctly, boosting intraclass compactness and interclass separability.

3.5 Binary Classification

In terms of classifying the net embeddings, SVM is used in conjunction with Principal Component Analysis (PCA). The use of PCA is needed primarily due to its proficiency in reducing the dimensionality of the feature space.

In the context of classification for high-dimensional image data, such as those extracted from the CNN net, the feature set can be overwhelmingly large. This high dimensionality not only poses computational challenges but can also lead to the phenomenon known as the "curse of dimensionality,"which potentially degrades the classifier's performance.

PCA addresses these issues by transforming the original, possibly correlated features into a set of linearly uncorrelated variables known as principal components. These components are ordered so that the first few retain most of the variation present in the original dataset. By selecting a subset of these components, PCA effectively reduces the data's dimensions while preserving the most critical variance characteristics.

Consequently, the PCA was adjusted for this study to retain 90% of the variance in the data, selecting a number of components that retain this large percentage of the total variation.

The main component in this block is the classifier, which categorizes images based on whether they contain *Leishmania* parasites, leveraging the extracted and reduced features. However, due to the inherent sensitivity of the estimator to data scaling, normalizing the features is crucial to enhancing both the performance and convergence of the model. Therefore, feature standardization was implemented by removing the mean and scaling to unit variance, ensuring that each feature contributes equally to the classification.

4

Experimental Results and Discussions

This chapter covers an in-depth study and integration of two separate microscopy image datasets from bone marrow aspirates. The acquisition, preprocessing, and annotation of these datasets merged, which are used to validate experimental results, are thoroughly addressed. The approaches for data splitting, and architectural complexity of the implemented CNN are detailed in the following sections.

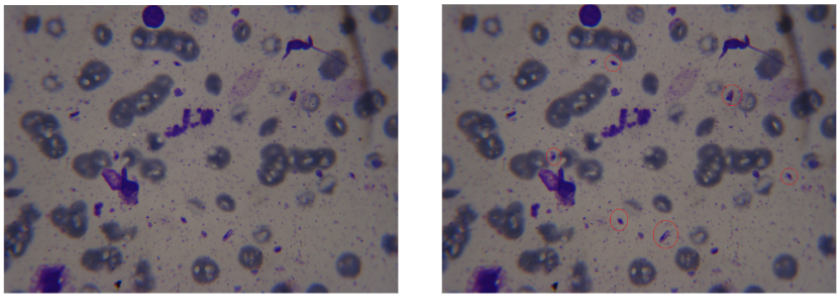
In addition, an investigation into metric learning losses and their respective parameters, as well as an exhaustive search for suitable SVM values, is conducted. The chapter concludes with a discussion regarding the presentation of the result, which includes several performance indicators like precision, recall, accuracy, F1-score, and the Matthew Correlation Coefficient (MCC), as well as ROC/AUC curve analysis and embedding space graphic visualization.

4.1 Data Acquisition

To conduct the experiments, a dataset was constructed from images sourced from two distinct collections. The first dataset was gathered by Farahi et al. (2014) and consists of 45 pairs of color microscope images of bone marrow aspirates, captured using a digital camera (Sony DSC H9) attached to an optical microscope (Olympus-CH40RF200) [1]. Figure 11 exemplifies a pair of images from this database.

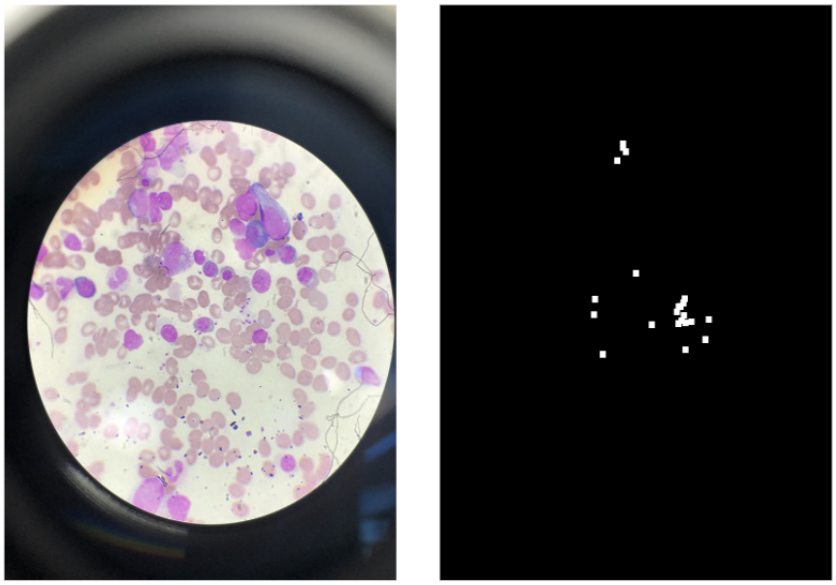
The second database, created by Marinho (2020), comprises 68 pairs of images captured using a mobile phone (iPhone 8) attached to an optical microscope with a 1000x magnification. Each pair, as depicted by Figure 12, consists of a color image and a corresponding binary mask (black and white), with the same dimensions as the original image, where the white regions indicate the location of parasites in the RGB image.

[1] Available at https://sites.google.com/site/hosseinrabbanikhorasgani/available-datasets/dataset-of-leishmania-parasite-in-microscopic-images



(a) Image from microscope (b) Same image marked

Figura 11: Example of data from Fahari Dataset (Dataset 1)



(a) Image from microscope (b) Binary mask

Figura 12: Example of data from Marinho Dataset (Dataset 2)

4.1.1 Binary Masks Generation and ROI Segmentation

As observed in Figure 12a, the external area of the microscope is visible in the images. Given that the employed method is based on patch analysis, maintaining the peripheral area would generate numerous inconsiderable patches, slowing down the algorithm and degrading the overall classification performance. To address this, Lisboa (2023) considered utilizing the Hough Circles algorithm to identify the microscope's encompassing circle, creating a binary image to isolate this region, and finally extracting the minimal bounding square for precise ROI segmentation.

In Dataset 1, additional preprocessing was performed to deal with the images with parasite markings. From these, binary masks were generated corresponding to the RGB images, similar

to Figure 12b. These images were manually labeled using the computer program `Photoshop`[2].

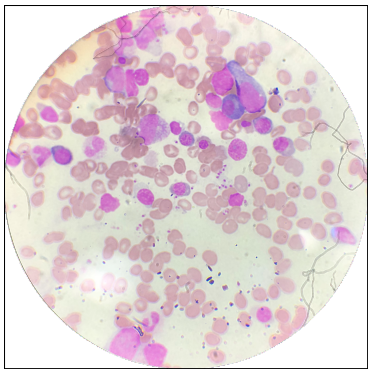


Figura 13: Example of Dataset 2 final images.

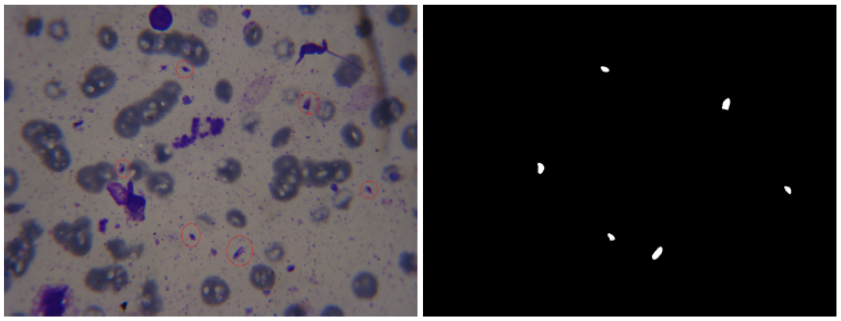


Figura 14: Before (left) and after (right) of manual data labelling.

4.2 Data Preprocessing

The Pillow library, a fork and successor of the Python Imaging Library (PIL), was utilized for automatic image enhancement to improve the visual quality of the images. Specifically, the method `PIL.Contrast.Enhance` was utilized, where the contrast of the images is adjusted using a hyper-parameter known as "factor".

According to the Pillow documentation[3], this hyper-parameter is a floating point value controlling the enhancement: a factor value of 1.0 returns a copy of the original image, whereas higher values increase the contrast of the image.

The selection of the hyper-parameter value at 1.5 was determined through an iterative process based on visual observation of the resultant images. A series of experiments were conducted, varying the contrast enhancement factor, and the resulting images were visually inspected

[2]Available at https://www.adobe.com/products/photoshop.html
[3]Available at https://pillow.readthedocs.io/en/stable/reference/ImageEnhance.html

to assess the clarity and distinction of relevant features. The factor of 1.5 was chosen as it consistently produced images with improved clarity and contrast without exaggerating the features or introducing excessive noise, thus maintaining the integrity of the original image data.

4.3 Data Splitting

Having enhanced the RGB images (Section 3.1) and obtained their respective binary masks, the subsequent step involved merging both datasets. This was followed by executing the image clipping algorithm (Figure 10). Afterward, data augmentation for the positive class and downsampling for the negative class were conducted. These procedures resulted in a total of 65,202 images (96x96x3-shaped), ready for splitting, as delineated in Table 2.

	Original size	**Patches (positive-negative)**	**Synthetic Data**	**Randomly removed from negative class**
Dataset 1	45	997 - 24859	-	-
Dataset 2	68	3319 - 38714	-	-
Dataset 1+2	-	4316 - 63573	-	-
Dataset 1+2 (augmented and balanced)	-	21734 - 43468	17418	20105

Tabela 2: Quantity of data through data wrangling stages.

Training deep learning networks requires using different sets of data, including training, validation, and testing for maximal efficacy. The training set is used to educate the model, enabling it to learn from labeled examples and fine-tune its parameters. The validation set serves to monitor and tune the model's performance during training, providing feedback to optimize hyper-parameters and avoid overfitting. Finally, the test set is crucial for assessing the trained model's performance, offering an unbiased evaluation of its generalization to new, unseen data, and confirming its suitability for deployment in real-world applications. This tripartite division of data ensures a comprehensive evaluation of the model's predictive abilities and effectiveness.

For this, this experiment used 70% of the data for training, 15% for validation, and 15% for testing. Table 3 illustrates the division of patches in the database.

	Total	**Positive**	**Negative**
Training (70%)	45641	15142	30499
Validation (15%)	9780	3304	6476
Test (15%)	9781	3288	6493
Total	65202	21734	43468

Tabela 3: Quantity of patches for training, validation and, testing.

4.4 Basal Model Architecture and Hyper-parameters

The subsequent sections provide a detailed overview of the architecture and hyper-parameters that underpin the deep learning model, the next phase of the pipeline. This includes an in-depth exploration of the CNN used for feature extraction, along with the configurations of its layers.

It will also delve into the parameter values of the four loss functions analyzed in the deep metric learning process, elucidating how these parameters were optimized to enhance model performance. Additionally, the last section will cover how the parameters of the SVM classifier were chosen, detailing the exhaustive search process undertaken to identify the optimal settings for accurate and reliable classification.

4.4.1 Convolutional Neural Network

- Three 2D convolutional layers with increasing channel depths (32, 64, 128 respectively), each with a kernel size of 3 and stride of 1.
- Each convolutional layer is followed by a batch normalization layer corresponding to its channel depth, stabilizing the learning process by normalizing the output of each layer.
- After each convolutional and batch normalization layer, a Rectified Linear Unit (ReLU) activation function is applied, introducing non-linear properties to the model.
- Two max-pooling layers are used after the second and third convolutional layers, each with a window of 2x2, reducing the spatial dimensions of the feature maps.
- Two dropout layers (dropout1 with 0.25 rate, dropout2 with 0.5 rate) are included for regularization, reducing the chance of overfitting by randomly zeroing out neurons during training.
- Two linear layers (fc1 and fc2) for feature compression and transformation. fc1 reduces the dimensionality to 512, and fc2 maps these to the specified embedding size.

Additionally, the training phase was conducted with a batch size of 32 images and 100 epochs monitored by an early stopping mechanism; as a result, the precise value for the patience parameter was altered based on the observed performance of the loss function during training. It ranged from 13 to 20 epochs. A learning rate of 0.001 was initially set and then dynamically adjusted using a learning rate scheduler that reduced the rate by a factor of 0.1 if the validation loss reached a plateau. The output embedding size was established at 128.

4.4.2 Metric Learning Losses Parameters

The `Python Metric Learning`[4] package provides access to various DML loss functions that are readily accessible and seamlessly integrable with the core deep learning architecture and, therefore, broadly used in this research. Below, one may find discussions regarding the parameter value choices.

Triplet Loss

The *margin* hyper-parameter is set at a value of 0.3, providing a quantifiable boundary that ensures positive examples are closer to the anchor than the negative examples by at least this margin. The choice of the Cosine Similarity as the **distance** metric deviates from the conventional Euclidean distance, favoring the angular difference between the feature vectors, thus emphasizing the orientation rather than the magnitude of the vectors in the learning embeddings. In other words, by focusing on the angle, the model can recognize parasites based on their shape and structure.

Circle Loss

The relaxation factor that controls the radius of the decision boundary, *m* function parameter, was set to 0.4 since the Sun et al. (2020) uses 0.25 for face recognition and 0.4 for fine-grained image retrieval. Along the same lines, the authors use 256 as the value for the *gamma* parameter for face recognition and 80 for fine-grained image retrieval.

MultiSimilarity Loss

For the testing phase of the project, the hyper-parameters *alpha* and *beta* were set to 2 and 50. This configuration was intended to appropriately balance the contribution of each pair type to the loss, enhancing the model's focus on informative pairs. Moreover, the margin parameter *lambda* was established at a value of 1. This value determines the threshold at which pairs are considered either positively or negatively similar, thereby influencing the difficulty of the optimization problem.

NPairs Loss

No changes were performed. The results shown by this loss were obtained the with package's default configuration.

[4]Documentation available at https://kevinmusgrave.github.io/pytorch-metric-learning/

4.4.3 SVM Exhaustive Search

The SVM classifier was tuned using Grid Search with cross-validation. The hyper-parameters explored in the grid search include *C* (regularization parameter) with values [0.1, 1, 10], and *gamma* (kernel coefficient) with values [1, 0.1, 0.01, 0.001]. The search is conducted using the recall macro as the scoring metric and is executed in parallel to improve computational efficiency. This procedure seeks to determine the best SVM settings for each CNN trained with various loss functions, ultimately classifying their respective transformed embeddings efficiently with the highest recall possible. The best-performing SVM model and its corresponding score can be visualized in Table 4.

4.4.4 Machine Setup

The computational experiments presented in this study were run on a machine whose specifications are detailed below.

- `AMD Ryzen 7 5800H` processor with 8 cores and a frequency of 3.2GHz.
- 16GB of RAM.
- `NVIDIA GeForce RTX 3060` graphics, with 6GB GB of dedicated memory, as well as `CUDA` v11.3.1 and `cuDNN` v8.2.1 acceleration libraries.
- Operating system `Windows 11`.
- `Pytorch` deep learning framework v2.1.0.
- Code available at `https://github.com/yrribeiro/clf-leishmania`[5]

4.5 Results

4.5.1 SVM Grid Search Best Parameters

	C	Gamma	Recall Macro
Triplet	10	1	0.9915
Circle	10	1	**0.9994**
MultiSimilarity	10	1	0.9750
NPairs	10	1	0.9441

Tabela 4: SVM hyper-parameter values exhaustive search results for each loss function.

[5]Going open source after this work presentation.

4.5.2 Classification Metrics and MCC

The values in the tables 5 and 6 were obtained using a stratified cross-validation method on the test dataset. Five folds were used in this approach to ensure an adequate representation of the distribution of classes in the data, which was made possible by the implementation of `Scikit-Learn's 'StratifiedKFold'`.

All trained models were loaded and tested against the same data set. Predictions were made using the appropriate classifier on scaled embeddings, and metrics such as precision, recall, f1-score, and accuracy were calculated for each class. The averages and standard deviations of these metrics were then computed from the sum of all the folds' outputs, yielding the data displayed within the tables, and offering a detailed and representative examination of the model's overall performance.

	Precision		**Recall**		**F1-Score**		**Accuracy**	
	Mean	**STD**	**Mean**	**STD**	**Mean**	**STD**	**Mean**	**STD**
Triplet	**0.9635**	0.0042	0.8283	0.0115	0.8908	0.0076	0.9318	0.0044
Circle	**0.9872**	0.0024	**0.9830**	0.0046	**0.9850**	0.0020	**0.9900**	0.0013
Multisimilarity	0.8862	0.0167	**0.9613**	0.0057	**0.9221**	0.0084	0.9454	0.0064
NPairs	0.8941	0.0083	0.9504	0.0119	0.9213	0.0087	**0.9455**	0.0059

Tabela 5: Classification metrics report comparison - **Positive class**

	Precision		**Recall**		**F1-Score**		**Accuracy**	
	Mean	**STD**	**Mean**	**STD**	**Mean**	**STD**	**Mean**	**STD**
Triplet	0.9190	0.0050	**0.9841**	0.0018	0.9504	0.0031	0.9318	0.0044
Circle	**0.9914**	0.0023	**0.9935**	0.0012	**0.9925**	0.0010	**0.9900**	0.0013
Multisimilarity	**0.9796**	0.0029	0.9373	0.0104	0.9580	0.0051	0.9454	0.0064
NPairs	0.9741	0.0061	0.9430	0.0046	**0.9583**	0.0045	**0.9455**	0.0059

Tabela 6: Classification metrics report comparison - **Negative class**

The Matthews Correlation Coefficient (MCC) is a binary classification performance indicator that provides a fair assessment even when the classes are unequal in number (Chicco and Jurman, 2020). The MCC yields a result between -1 and 1, with 1 representing a perfect prediction, 0 expressing no better than a random guess, and -1 representing complete disagreement between forecast and reality.

MCC produces a high score only if the prediction performed well in all four confusion matrix categories, proportionally to the number of positive and negative items in the dataset. The MCC formula is as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

	MCC
Triplet	0.8467
Circle	**0.9775**
MultiSimilarity	0.8818
NPairs	0.8806

Tabela 7: Matthew Correlation Coefficient for all tested losses.

According to these data, one can infer that the overall performance of the CNNs modified versions was excellent, with the Circle Loss model demonstrating superior performance across all classification metrics. However, these findings also raise intriguing questions regarding the limitations of each loss function during the learning process.

While the Triplet loss function demonstrated a balanced proficiency in distinguishing negative examples, as evidenced by its high recall in the negative class, it also demonstrated a tendency toward higher false negatives in the positive class, as evidenced by the lower recall shown in Table 5. This also influences its MCC score of 0.8467 (Table 7), which, despite being the lowest of the examined functions, still suggests acceptable predictive quality. Being as good as a human specialist (Section 1.1).

On the other hand, MultiSimilarity stands as an improvement of Triplet and falls behind the Circle model. Despite possessing the lowest specificity, which suggests some challenges in correctly identifying all negative instances, its sensibility outperforms Triplet by 13%. A fair trade, since the nature of VL treatment is more tolerable to errors than missing a critical diagnosis. This is theoretically consistent with the MultiSimilarity function's objective of simultaneously pulling together similar examples and pushing apart dissimilar ones within the same batch, which may account for its relatively strong discriminative power.

In sequence, the loss function NPairs presented an outcome with a small difference from MultiSimilarity yet significant in the context of the research. Recalling what was previously detailed in Section 2.1.6, MultiSimilarity uses an iterative mining and weighting technique to identify informative pairs based on positive relative similarity and assigns higher weights to these pairs, considering self-similarity and negative relative similarity.

In this research context, the negative instances received higher weight (parameter beta), emphasizing class separation. NPairs, however, compare a positive instance against multiple negative instances simultaneously, encouraging the model to differentiate a given instance from several negative classes at the same time. As a result, the two took the approach of prioritizing the distinction between negative instances within the feature space, obtaining technically similar results in all quantitative analyses.

4.5.3 ROC/AUC Curve

When analyzing the Receiver Operating Characteristic (ROC) curve, we consider that a curve closer to the upper left corner indicates superior performance (a high percentage of true positives and a low rate of false positives). The Area Under the Curve (AUC) measures the model's ability to distinguish between positive and negative classes. An AUC of 1.0 suggests a flawless model that properly classifies all positives and negatives. An AUC of 0.5, on the other hand, shows that the performance is no better than chance.

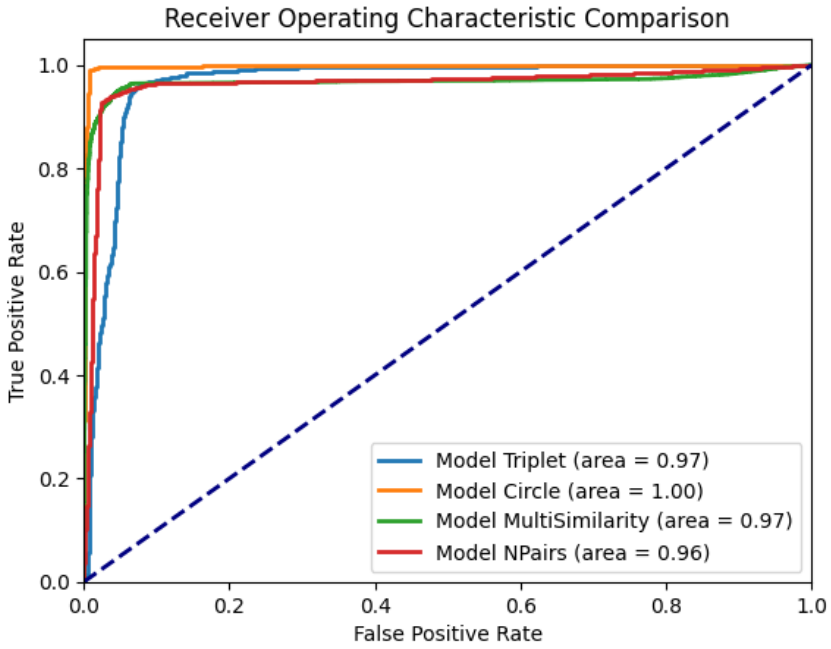


Figura 15: ROC/AUC curve comparison for all models tested.

Analysis of the ROC curves confirms that the Circle Model is superior in terms of classification performance, with AUC values approaching perfection, as evidenced by the mathematical rounding to 1.0. This indicates an almost ideal distinction between positive and negative classes. All the models evaluated exhibited AUCs greater than 95%, reflecting a robust discrimination capacity that substantially exceeds randomness, denoted by the AUC baseline (blue dashed line).

The Triplet and NPairs models, although effective, showed lower sensitivity, implying an increased propensity for false positives when compared to the Circle Model. On the other hand, the MultiSimilarity Model, despite its high AUC, showed the lowest specificity among the models tested, suggesting a higher probability of incorrectly classifying negatives as positives. These conclusions, anchored in the quantitative AUC metrics, reinforce the overall competence of the models in the binary classification task.

4.5.4 Embedding Space Visualization

The t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm is a dimensionality reduction approach that is particularly well-suited to visualizing high-dimensional data sets. Thus it is adopted here to demonstrate how models map data characteristics onto an embedded space and how this affects the ability to separate classes[6]. By visualizing the data in this way, it is possible to intuitively base the model's performance.

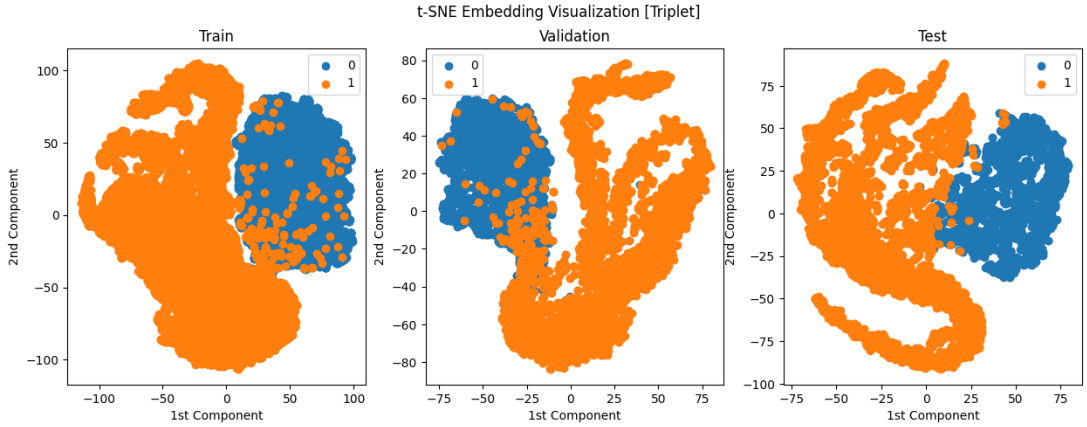


Figura 16: t-SNE Embedding visualization [Triplet].

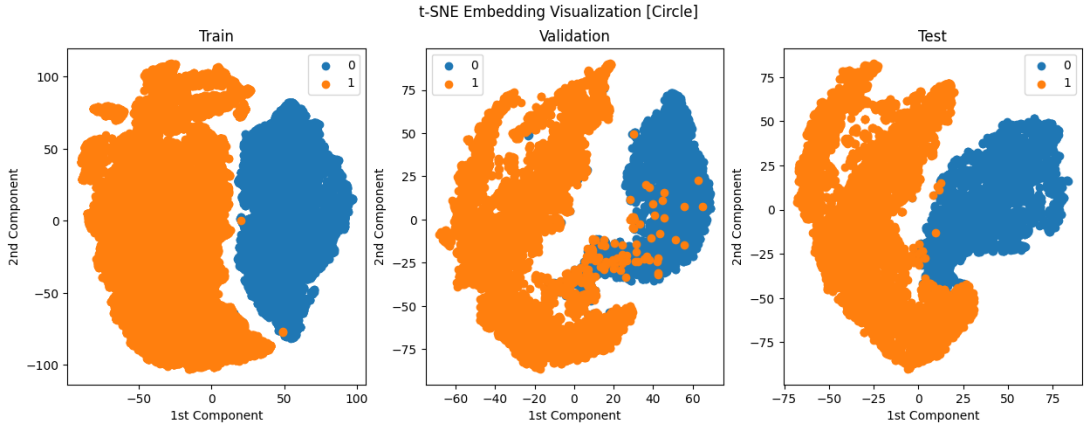


Figura 17: t-SNE Embedding visualization [Circle].

In Figure 16, Triplet demonstrated a reasonable distinction of classes, resulting in dense clusters but with overlapping areas where the model may be more likely to commit errors. It is also possible to notice an inconsistency between the validation data and the other sets. This could be caused either by overfitting issues or by the model having difficulty defining the discriminative characteristics in the validation set due to data variance.

[6]Class 0 indicates the presence of VL amastigote, class 1 the absence.

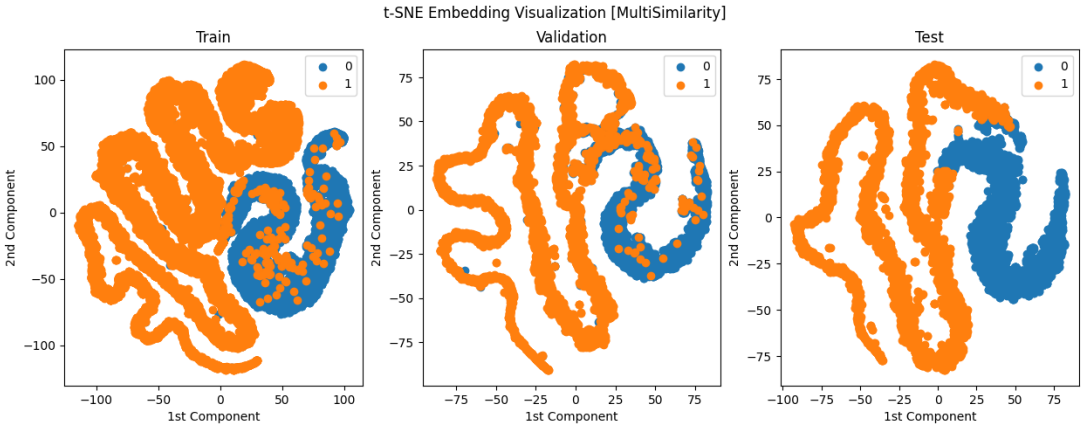


Figura 18: t-SNE Embedding visualization [MultiSimilarity].

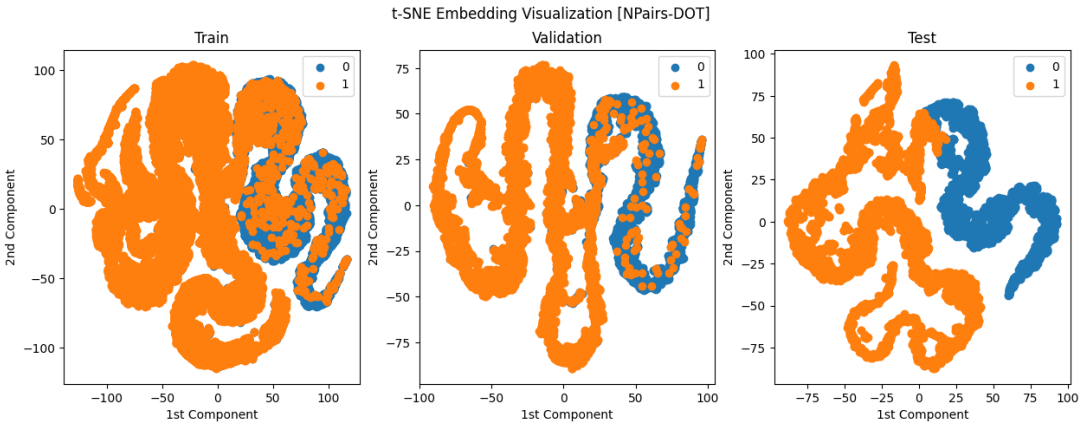


Figura 19: t-SNE Embedding visualization [NPairs].

As measures against overfitting were adopted and other models performed consistently well, it leads one to believe that this result was due to the nature of the function itself not being able to generalize the data accurately.

Circle Loss's flexibility, which allows it to focus on challenging pairs and draw an optimal margin between classes, may have contributed to its exceptional results. This performance outcome can also be seen in Figure 17, where the training and test categories clusters appear to be well separated and consistent through all data sets.

When compared to the tightly spaced embeddings of the Circle Loss model, the MultiSimilarity embeddings, shown in Figure 18, demonstrate a less compact but still visible separation of classes. Given that we are analyzing images of bone marrow aspirates, which can contain thousands of different biological components (not just *Leishmania*), the pattern of high variance in negative class data is justified. This is simply because this loss function is designed to take into account multiple similarities and divergences within the same batch of data, making it particularly sensitive to intraclass variance and more useful than other functions in scenarios where this variety is relevant.

4.5.5 Visual Analysis of the Classification

This section provides valuable visual information to understand in which cases and why the model fails at classifying amastigotes. Each model classified the same four batches, comprising 32 patches per batch. The findings are displayed in the image titles, paired with the true categories. In addition, the number and types of errors made for these image sets are reported. Figures 20 and 21 serve as examples where all models correctly identified positive and negative patches, respectively.

Figura 20: Example of **true positive** images.

Figura 21: Example of **true negative** images.

Triplet

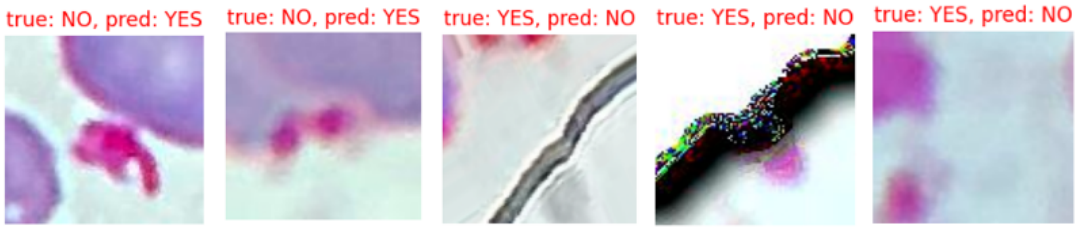


Figura 22: Example of **misclassified images** [Triplet].

Misclassified 13 out of 128 images, which corresponds to approximately 10% of the images. This included 5 false positives and 8 false negatives.

Circle

During this specific analysis, the CNN model trained with Circle Loss did not commit any errors in its classifications.

MultiSimilarity

Figura 23: Example of **misclassified images** [MultiSimilarity].

Misclassified 6 out of 128 images, approximately 5% of the images. This included 4 false positives and 2 false negatives.

NPairs

Figura 24: Example of **misclassified images** [NPairs].

Misclassified 11 out of 128 images, approximately 9% of the images. This included 8 false positives and 3 false negatives.

Analyzing the false positives reported in Figures 22, 23 and 24 and considering that all target images used for training introduce, roughly, an oval shape with a full circle in the middle, it has become clear that most of the FP cases occur due to other biological structures resembling *Leishmania*. Some structures also share a color intensity similarity, making the distinction even harder. Likewise, parasites pictured overlapped by other cell structures may hinder the model from correctly classifying the positive instances, as such conditions were scarce in the training set.

Important to notice that, a few amount of amastigotes were cropped by image borders but still classified as *Leishmania* positive by the clipping algorithm. This phenomenon is affected

by α, the parasite area threshold in the clipping algorithm. This visual analysis was instrumental in establishing a threshold value that results in the minimum number of parasites affected by this condition.

4.6 Performance Comparison

Method	Detection/Classification		Segmentation		Number of Images
	Metric	Result	Metric	Result	
Isaza-Jaimes et al.	Sensitivity	0.787			*45
Coelho et al.			ACC	0.95	-
Górriz et al.			Recall	0.823	45
			Precision	0.757	
			F1-score	0.777	
			Dice	0.777	
Gonçalves et al.			ACC	0.991	78
			Sensitivity	0.722	
			Specificity	0.996	
			AUC	0.859	
			Dice	0.804	
Salazar et al.			Dice	0.85	*45
Proposed method	Sensitivity	0.983 (0.0046)	-	-	*113
	Specificity	0.993 (0.0012)			
	ACC	0.990 (0.0013)			
	MCC	0.977			
	AUC	0.999			

Tabela 8: The proposed method's performance in comparison to the state of the art. In the last column, works that used the same dataset (Fahari Dataset) are noted with an asterisk.

In terms of hardware efficiency and computational resource utilization, the top-performing model, Circle, showcased a convergence time of 58 minutes, with each epoch processed in an average of 93 seconds. This performance was achieved through optimized CPU usage, recorded at 3.65%. Additionally, the model utilized 6.74 GB of RAM, amounting to 56% of the total available RAM.

Notably, as measured by the `GPUtil` library, the GPU utilization on average was 57% of its total processing capacity, and 44% of the GPU's memory was employed. During grid search operations, the model took 46 seconds to complete 60 fits, demonstrating effective parameter optimization. When assessing the entire test set, the classification process was completed in 8.6 seconds, reflecting the model's responsiveness in practical applications.

For the other three models (Triplet, MultiSimilarity, and NPairs), the average convergence time was 49 minutes (±10.2 min), with each epoch taking 105.31 seconds (± 5.75s). The average

CPU and GPU utilizations were 1.33% (± 2.72) and 52% (± 4%) respectively, with an average RAM usage of 10.56 GB (± 0.23GB).

The grid search process for these models was completed in an average time of 1.10 minutes, with a variation of ± 2.25 minutes. When classifying the entire test batch, the models took an average of 54 seconds, with a standard deviation of ± 0.75 seconds. It's important to note that the size of each model is approximately 121 MB, which represents a balance between the complexity of the models and their storage efficiency.

5

Conclusion

The comparison of various deep metric learning methods has shown the significant potential of the evaluated models for applications in cytological data imaging. It revealed Circle Loss as the standout performer, excelling across all classification metrics, especially sensitivity (98.3%) and specificity (99.3%), aligning well with the study's context.

The primary objective was to experimentally accentuate relevant areas in images and segment them into smaller patches for improved feature discernibility, a technique that was accomplished and most likely led to the models' improved performance. This performance was also influenced by the appropriate configuration of the SVM algorithm, which converted the deep metric learning models' learned characteristics into actionable diagnostic insights.

However, this evaluation also identified certain limits and topics for additional research. For instance, the Triplet loss function, while effective in certain aspects, demonstrated a tendency toward higher false negatives, indicating that additional preprocessing for image background differentiation and model fine-tuning could be useful to further reduce the false negative rate.

In summary, the research was successful in meeting its goals by building and assessing multiple deep metric learning algorithms, with Circle Loss emerging as especially valuable. The findings not only provide useful insights into the application of these models for medical diagnosis, but they also identify areas for future research to further develop these approaches.

5.1 Future Work

Based on the positive findings of this work, numerous future research directions have been identified to improve the use and effectiveness of the generated models in the field of parasitological diagnostics. These are some examples:

- Analyze image processing approaches for blurring or removing background structures in images to decrease false positives by minimizing the impact of components similar to the target parasites.

- Fine-tuning models with a more diverse range of *Leishmania* images to reduce the rate of false negatives.
- Explore the feasibility of incorporating the most effective models into existing VL diagnostic systems, offering a significant advancement in the field.
- Extend the experiment to other similar diseases, like malaria or Chagas disease, to evaluate the models' applicability and effectiveness in diagnosing a broader range of parasitic infections.

References

Akhoundi, M., Downing, T., Votýpka, J., Kuhls, K., Lukeš, J., Cannet, A., Ravel, C., Marty, P., Delaunay, P., Kasbari, M., Granouillac, B., Gradoni, L., and Sereno, D. (2017). Leishmania infections: Molecular targets and diagnosis. *Molecular Aspects of Medicine*, 57:1–29. Leishmania Infections: Molecular Targets and Diagnosis.

Antinori, S., Calattini, S., Longhi, E., Bestetti, G., Piolini, R., Magni, C., Orlando, G., Gramiccia, M., Acquaviva, V., Foschi, A., Corvasce, S., Colomba, C., Titone, L., Parravicini, C., Cascio, A., and Corbellino, M. (2007). Clinical Use of Polymerase Chain Reaction Performed on Peripheral Blood and Bone Marrow Samples for the Diagnosis and Monitoring of Visceral Leishmaniasis in HIV-Infected and HIV-Uninfected Patients: A Single-Center, 8-Year Experience in Italy and Review of the Literature. *Clinical Infectious Diseases*, 44(12):1602–1610.

Bakator, M. and Radosav, D. (2018). Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction*, 2(3):47.

Butt, U. M., Letchmunan, S., Ali, M., Hassan, F. H., Baqir, A., Sherazi, H. H. R., et al. (2021). Machine learning based diabetes classification and prediction for healthcare applications. *Journal of healthcare engineering*, 2021.

Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13.

Coelho, G., Galvão Filho, A. R., Viana-de Carvalho, R., Teodoro-Laureano, G., Almeida-da Silveira, S., Eleutério-da Silva, C., Pereira, R. M. P., Soares, A. d. S., Soares, T. W. d. L., Gomes-da Silva, A., Napolitano, H. B., and Coelho, C. J. (2020). Microscopic image segmentation to quantification of leishmania infection in macrophages. *Fronteiras: Journal of Social, Technological and Environmental Science*, 9(1):488–498.

Dodge, S. and Karam, L. (2016). Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE.

Elmahallawy, E. K., Sampedro, A. M., Rodriguez-Granger, J., Hoyos-Mallecot, Y., Agil, A., Navarro, J. M., and Fernández, J. (2014). Diagnosis of leishmaniasis. *Journal of Infection in Developing Countries*, 8(8):961 – 972.

Erber, A. C., Sandler, P. J., de Avelar, D. M., Swoboda, I., Cota, G., and Walochnik, J. (2022). Diagnosis of visceral and cutaneous leishmaniasis using loop-mediated isothermal amplification (lamp) protocols: a systematic review and meta-analysis. *Parasites & Vectors*, 15(1):34.

Farahi, M., Rabbani, H., and Talebi, A. (2014). Automatic boundary extraction of leishman bodies in bone marrow samples from patients with visceral leishmaniasis. *Journal of Isfahan Medical School*, 32(286):726–739.

Farahi, M., Rabbani, H., Talebi, A., Sarrafzadeh, O., and Ensafi, S. (2015). Automatic segmentation of Leishmania parasite in microscopic images using a modified CV level set method. In Wang, Y. and Jiang, X., editors, *Seventh International Conference on Graphic and Image Processing (ICGIP 2015)*, volume 9817, page 98170K. International Society for Optics and Photonics, SPIE.

Fuhad, K. M. F., Tuba, J. F., Sarker, M. R. A., Momen, S., Mohammed, N., and Rahman, T. (2020). Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application. *Diagnostics*, 10(5).

Gonçalves, C., Borges, A., Dias, V., Marques, J., Aguiar, B., Costa, C., and Silva, R. (2023). Detection of human visceral leishmaniasis parasites in microscopy images from bone marrow parasitological examination. *Applied Sciences*, 13(14).

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Górriz, M., Aparicio, A., Raventós, B., Vilaplana, V., Sayrol, E., and López-Codina, D. (2018). Leishmaniasis parasite segmentation and classification using deep learning. In Perales, F. J. and Kittler, J., editors, *Articulated Motion and Deformable Objects*, pages 53–62, Cham. Springer International Publishing.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Hu, J., Lu, J., and Tan, Y.-P. (2015). Deep metric learning for visual tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(11):2056–2068.

Isaza-Jaimes, A., Bermúdez, V., Bravo, A., Castrillo, J. S., Lalinde, J. D. H., Fossi, C. A., Flórez, A., and Rodríguez, J. E. (2021). A computational approach for Leishmania genus protozoa detection in bone marrow samples from patients with visceral Leishmaniasis.

Kaya, M. and Bilge, H. Ş. (2019). Deep metric learning: A survey. *Symmetry*, 11(9):1066.

Kelleher, J. D. (2019). *Deep learning*. MIT press.

Koirala, A., Jha, M., Bodapati, S., Mishra, A., Chetty, G., Sahu, P. K., Mohanty, S., Padhan, T. K., Mattoo, J., and Hukkoo, A. (2022). Deep learning for real-time malaria parasite detection and counting using yolo-mp. *IEEE Access*, 10:102157–102172.

Kumari, D., Perveen, S., Sharma, R., and Singh, K. (2021). Advancement in leishmaniasis diagnosis and therapeutics: An update. *European Journal of Pharmacology*, 910:174436.

Lisboa, E. A. d. L. (2023). Avaliação de métodos clássicos de detecção de características no auxílio à identificação de amastigotas de leishmaniose. *Repositório Institucional da UFAL*.

Liu, Y., Du, J., Vong, C.-M., Yue, G., Yu, J., Wang, Y., Lei, B., and Wang, T. (2022). Scale-adaptive super-feature based metricunet for brain tumor segmentation. *Biomedical Signal Processing and Control*, 73:103442.

Liu, Z., Jin, L., Chen, J., Fang, Q., Ablameyko, S., Yin, Z., and Xu, Y. (2021). A survey on applications of deep learning in microscopy image analysis. *Computers in Biology and Medicine*, 134:104523.

Lu, J., Hu, J., and Zhou, J. (2017). Deep metric learning for visual understanding: An overview of recent advances. *IEEE Signal Processing Magazine*, 34(6):76–84.

Marinho, T. T. (2020). Aspectos clínicos laboratoriais e o uso da modelagem computacional no diagnóstico da leishmaniose visceral. *Repositório Institucional da UFAL*.

Ni, J., Liu, J., Zhang, C., Ye, D., and Ma, Z. (2017). Fine-grained patient similarity measuring using deep metric learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1189–1198.

Pak, M. and Kim, S. (2017). A review of deep learning in image recognition. In *2017 4th international conference on computer applications and information processing technology (CAIPT)*, pages 1–3. IEEE.

Peters, B. S., Fish, D., Golden, R., Evans, D. A., Bryceson, A. D. M., and Pinching, A. J. (1990). Visceral Leishmaniasis in HIV Infection and AIDS: Clinical Features and Response to Therapy. *QJM: An International Journal of Medicine*, 77(2):1101–1111.

Qin, S. (2022). Fast image scanning microscopy with efficient image reconstruction. *OPTICS AND LASERS IN ENGINEERING*, 151.

Reimão, J. Q., Coser, E. M., Lee, M. R., and Coelho, A. C. (2020). Laboratory diagnosis of cutaneous and visceral leishmaniasis: Current and future methods. *Microorganisms*, 8(11).

Ritter, F., Boskamp, T., Homeyer, A., Laue, H., Schwier, M., Link, F., and Peitgen, H.-O. (2011). Medical image analysis. *IEEE pulse*, 2(6):60–70.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Salazar, J., Vera, M., Huérfano, Y., Vera, M. I., Gelvez-Almeida, E., and Valbuena, O. (2019). Semi-automatic detection of the evolutionary forms of visceral leishmaniasis in microscopic blood smears. *Journal of Physics: Conference Series*, 1386(1):012135.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Soberanis-Mukul, R., Uc-Cetina, V., Brito-Loeza, C., and Ruiz-Piña, H. (2013). An automatic algorithm for the detection of trypanosoma cruzi parasites in blood sample images. *Computer Methods and Programs in Biomedicine*, 112(3):633 – 639.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Spilger, R., Lee, J.-Y., Chagin, V. O., Schermelleh, L., Cardoso, M. C., Bartenschlager, R., and Rohr, K. (2021). Deep probabilistic tracking of particles in fluorescence microscopy images. *MEDICAL IMAGE ANALYSIS*, 72.

Srivastava, P., Dayama, A., Mehrotra, S., and Sundar, S. (2011). Diagnosis of visceral leishmaniasis. *Transactions of The Royal Society of Tropical Medicine and Hygiene*, 105(1):1–6.

Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., and Wei, Y. (2020). Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6398–6407.

Sundgaard, J. V., Harte, J., Bray, P., Laugesen, S., Kamide, Y., Tanaka, C., Paulsen, R. R., and Christensen, A. N. (2021). Deep metric learning for otitis media classification. *Medical Image Analysis*, 71:102034.

van Griensven, J. and Diro, E. (2019). Visceral leishmaniasis: Recent advances in diagnostics and treatment regimens. *Infectious Disease Clinics of North America*, 33(1):79–99.

Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R. (2019). Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5022–5030.

Werneck, G., Rodrigues, L., Santos, M., Araújo, I., Moura, L., Lima, S., Gomes, R., Maguire, J., and Costa, C. (2002). The burden of leishmania chagasi infection during an urban outbreak of visceral leishmaniasis in brazil. *Acta Tropica*, 83(1):13 – 18.

WHO TEAM, C. o. N. T. D. (2023). Operational manual on leishmaniasis vector control, surveillance, monitoring and evaluation. page 119.

WHO TEAM, O. o. L. and Services, H. L. (2023). Status of endemicity of visceral leishmaniasis: 2022.

Yang, F., Poostchi, M., Yu, H., Zhou, Z., Silamut, K., Yu, J., Maude, R. J., Jaeger, S., and Antani, S. (2020). Deep learning for smartphone-based malaria parasite detection in thick blood smears. *IEEE Journal of Biomedical and Health Informatics*, 24(5):1427–1438.

Yang, Z., Benhabiles, H., Windal, F., Follet, J., Leniere, A.-C., and Collard, D. (2022). A coarse-to-fine segmentation methodology based on deep networks for automated analysis of cryptosporidium parasite from fluorescence microscopic images. In Huo, Y., Millis, B., Zhou, Y., Wang, X., Harrison, A., and Xu, Z., editors, *MEDICAL OPTICAL IMAGING AND VIRTUAL MICROSCOPY IMAGE ANALYSIS, MOVI 2022*, volume 13578 of *Lecture Notes in Computer Science*, pages 156–166. 1st International Workshop on Medical Optical Imaging and Virtual Microscopy Image Analysis (MOVI), Singapore, SINGAPORE, SEP 18, 2022.

Yaseen, Q. et al. (2021). Spam email detection using deep learning techniques. *Procedia Computer Science*, 184:853–858.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Deep metric learning for person re-identification. In *2014 22nd international conference on pattern recognition*, pages 34–39. IEEE.

Zhang, C., Jiang, H., Jiang, H., Xi, H., Chen, B., Liu, Y., Juhas, M., Li, J., and Zhang, Y. (2022). Deep learning for microscopic examination of protozoan parasites. *Computational and Structural Biotechnology Journal*, 20:1036–1043.

Zhong, A., Li, X., Wu, D., Ren, H., Kim, K., Kim, Y., Buch, V., Neumark, N., Bizzo, B., Tak, W. Y., et al. (2021). Deep metric learning-based image retrieval system for chest radiograph and its clinical applications in covid-19. *Medical Image Analysis*, 70:101993.