Trabalho de Conclusão de Curso

Microscopy Image Analysis Using Deep Metric Learning: A Study on the Detection and Classification of Leishmania Amastigotes and Other Parasites

Carllos Eduardo Ferreira Lopes
cefl@ic.ufal.br

Orientadora:

Profa. Dra. Fabiane da Silva Queiroz

Maceió, 25 de Outubro de 2024

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Helena Cristina Pimentel do Vale – CRB4/ 661

L864m Lopes, Carllos Eduardo Ferreira.
Microscopy image analysis using deep metric learning : a study on the detection and classification of leishmania amastigotes and other parasites / Carllos Eduardo Ferreira Lopes. – 2025.
53 f.: il.

Orientadora: Fabiane da Silva Queiroz.
Monografia (Trabalho de Conclusão de Curso em Ciências da Computação) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2024.

Texto em Inglês.
Bibliografia: f. 49-53.

1. Deteção de parasitas. 2. Leishmaniose visceral. 3. Aprendizagem profunda. 4. Aprendizagem de métrica profunda. 5. Redes neurais convolucionais, I. Título.

CDU: 004.8

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Instituto de Computação – IC
Campus A. C. Simões - Av. Lourival de Melo Mota, BL 12
Tabuleiro do Martins, Maceió/AL - CEP: 57.072-970
Telefone: (082) 3214-1401

Trabalho de Conclusão de Curso - TCC

Formulário de Avaliação

Nome do Aluno
Carllos Eduardo Ferreira Lopes

Nº de Matrícula
19111150

Título do TCC (Tema)
Microscopy Image Analysis Using Deep Metric Learning: A Study on the Detection and Classification of Leishmania Amastigotes and Other Parasites

Banca Examinadora

Fabiane da Silva Queiroz
Orientadora
_____
Assinatura

Raquel da Silva Cabral
_____
Assinatura

Rodolfo Carneiro Cavalcante
_____
Assinatura

Data da Defesa
07/12/2024

Nota Obtida
10,00 (dez)

Observações

Coordenador do Curso
De Acordo

_____
Assinatura

Carllos Eduardo Ferreira Lopes

Microscopy Image Analysis Using Deep Metric Learning: A Study on the Detection and Classification of Leishmania Amastigotes and Other Parasites

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, 25 de Outubro de 2024

Agradecimentos

Agradeço primeiramente aos meus pais e à minha família, sabendo que qualquer agradecimento, ao ser posto em palavras, nunca será suficiente para demonstrar a gratidão que sinto no peito.

Obrigado, mãe, por ser minha guia e meu apoio incondicional durante toda a minha jornada.

Obrigado, pai, por ter me ensinado desde cedo o valor do trabalho duro e do estudo.

Obrigado, irmã, pelas palavras trocadas e pela torcida.

Obrigado, tia, pelo apoio; nunca esquecerei.

Obrigado, madrinha, por todo o cuidado.

Aos meus irmãos de vida, Gabriel, Matheus e Jucelio, obrigado pela parceria e por me mostrarem que sonhar alto é necessário.

Agradeço também a todos os amigos que fiz durante meu caminho; sem vocês, tudo teria sido mais difícil.

Resumo

A Leishmaniose Visceral, uma forma grave da doença causada pelo parasita *Leishmania donovani*, é fatal em mais de 95% dos casos não tratados e afeta principalmente pessoas de baixa renda com acesso limitado a cuidados de saúde. O diagnóstico padrão envolve a identificação de amastigotas do parasita, que são pequenas e difíceis de encontrar, tornando o exame uma tarefa desafiadora que requer habilidade.

Para ajudar os profissionais de saúde, este estudo propõe uma nova abordagem que combina aprendizagem métrica profunda com classificação supervisionada para a detecção rápida da leishmaniose visceral. A metodologia divide as imagens em pequenos fragmentos (*patches*) para melhorar a avaliação de quatro funções de perda, que ajudam uma Máquina de Vetores de Suporte (SVM) a diagnosticar a doença.

Os resultados mostraram que a função Circle teve o melhor desempenho, com 98,3% de sensibilidade e 99,3% de especificidade. Além da leishmaniose, exploramos o desempenho em outras infecções parasitárias, como Babesia, Toxoplasma, Trypanosoma, Plasmodium e Schistosoma, que também apresentaram resultados impressionantes, com alta precisão e sensibilidade. Essa abordagem sugere que a inteligência artificial pode ser uma ferramenta valiosa para melhorar o diagnóstico de doenças tropicais negligenciadas, tornando-o mais acessível e eficiente.

Palavras-chave: Deteção de Parasitas, Leishmaniose Visceral, Aprendizagem Métrica Profunda, Aprendizagem Profunda, Redes Neurais Convolucionais, Classificação Binária, Diagnóstico Automático de Doenças, Classificação Multiclasse, Babesia, Toxoplasma, Trypanosoma, Plasmodium e Shistosoma.

Abstract

Visceral Leishmaniasis, a severe form of the disease caused by the parasite Leishmania donovani, is fatal in over 95% of untreated cases and primarily affects low-income individuals with limited access to healthcare. The standard diagnosis involves identifying the parasite's amastigotes, which are small and difficult to find, making the examination a challenging task that requires skill.

To assist healthcare professionals, this study proposes a new approach that combines deep metric learning with supervised classification for the rapid detection of visceral leishmaniasis. The methodology divides images into small fragments (patches) to enhance the evaluation of four loss functions, which help a Support Vector Machine (SVM) diagnose the disease.

The results showed that the Circle loss function performed best, with 98.3% sensitivity and 99.3% specificity. In addition to leishmaniasis, we explored the performance on other parasitic infections, such as Babesia, Toxoplasma, Trypanosoma, Plasmodium, and Schistosoma, which also demonstrated impressive results, with high precision and sensitivity. This approach suggests that artificial intelligence can be a valuable tool for improving the diagnosis of neglected tropical diseases, making it more accessible and efficient.

Key-words: Parasite Detection, Visceral Leishmaniasis, Deep Metric Learning, Deep Learning, Convolutional Neural Networks, Binary Classification, Automated Disease Diagnosis, Multiclass Classification, Babesia, Toxoplasma, Trypanosoma, Plasmodium, and Schistosoma.

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
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
ROI	Region of Interest
ReLU	Rectified Linear Unit
DOT	Dot Product Similarity
COS	Cosine Similarity
KNN	K-Nearest Neighbors

Conteúdo

1

Introduction

1.1 Human Visceral Leishmaniasis

Leishmaniasis is a widespread and contagious disease caused by microscopic parasites called *Leishmania*. These single-celled organisms live inside human cells, making the disease difficult to treat. Leishmaniasis primarily affects people in low-resource settings and those with limited access to healthcare.

A particularly dangerous form of Leishmaniasis, known as visceral leishmaniasis (VL) or kala-azar, is caused by a group of Leishmania parasites called the *Leishmania donovani* complex. This form can be fatal if left untreated.

Untreated Visceral VL is highly fatal, with over 95% of patients succumbing to the disease if left untreated. The World Health Organization (WHO) [1] describes the symptoms of VL as irregular fever, weight loss, enlarged spleen and liver anemia. Leishmaniasis is affecting 99 countries and territories globally. Of these, 81 countries are considered endemic for VL, meaning the disease is constantly present in the region. In the Americas, VL is found in 12 countries, with South American nations like Brazil, Argentina, Colombia, Paraguay, and Venezuela bearing a significant burden of the disease.

Leishmaniasis is a spreading threat, with cases increasing in Central America. Honduras and Guatemala, previously reporting few cases, saw a significant rise in 2022 (52). In Southern Europe, the disease is a major risk for those with compromised immune systems, particularly individuals with AIDS (38).

Globally, the disease burden is concentrated in Brazil, East Africa, and India. This puts over a billion people living in endemic areas at risk. Although an estimated 50,000 to 90,000 new VL cases occur annually, only a fraction (25-45%) of those are reported to the WHO.

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

Diagnosis of Visceral Leishmaniasis (VL) involves both DNA and non-DNA based methods (6): DNA-based methods like PCR and qPCR offer high accuracy but are complex and expensive. These are primarily limited to research facilities and specialized hospitals in regions where VL is widespread (7; 24). Non-DNA-based methods, including serological tests and parasitological procedures, are more accessible but may lack accuracy, particularly for detecting asymptomatic cases (51; 24). These tests often look for antibodies or antigens produced by the body in response to the infection.

Microscopic examination of parasite tissues remains the most reliable method for diagnosing visceral leishmaniasis (VL), according to (12). This technique involves directly searching for the amastigote form, a specific stage of the Leishmania parasite, in samples like bone marrow, lymph nodes, or spleen obtained through aspiration or biopsy (49). Preparing smears for examination is simple and cost-effective, making it the preferred method in areas with limited resources where PCR technology might not be readily available (11).

Leishmania amastigotes, are microscopic parasites living inside human cells (39). These tiny, round or oval-shaped organisms, measuring only 2-4 micrometers in diameter, have distinct internal structures, including a nucleus and a kinetoplast]. Although microscopic examination is the traditional method for diagnosing VL, it can be challenging due to several factors: Identifying these minuscule parasites requires expertise and can be a lengthy and tiring process, as noted by (47). Difficulties in differentiation: Because they can resemble other elements within the sample, accurately distinguishing Leishmania amastigotes can be difficult for even trained professionals . These limitations highlight the need for alternative diagnostic methods that are faster, more reliable, and less dependent on expert interpretation.

While directly examining tissue microscopically offers a way to diagnose VL, its accuracy is limited, often missing cases even when the parasite is present. A more reliable approach involves taking a bone marrow biopsy and staining it with Giemsa [2], a common diagnostic stain. However, even this method only has a sensitivity of 60-85%, highlighting the ongoing challenge of accurately diagnosing VL (11).

Machine learning (ML) is transforming the way we diagnose diseases, including VL. ML automates repetitive tasks, freeing up healthcare professionals' time for more complex aspects of patient care. By removing human subjectivity, ML can reduce variability in diagnoses, leading to more reliable results. It can analyze large datasets of medical images much faster than human experts, allowing for quicker diagnoses. Also, algorithms can identify subtle patterns in medical images that might be missed by the human eye, leading to more accurate diagnoses. Specifically, computer vision and deep learning techniques have shown great promise in detecting VL in humans by analyzing bone marrow microscopy images with high precision.

[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 aims to investigate the potential of deep metric learning techniques, combined with Principal Component Analysis (PCA), in accurately diagnosing visceral leishmaniasis (VL) using microscopic images. The study focuses on three key objectives:

First, the study will assess the performance of deep learning models in identifying and classifying images containing VL parasites. Different models will be compared, from Convolutional Neural Networks to deep metric learning models used together with classic classifiers such as Support Vector Machines and K-Nearest Neighbors.

Second, we will compare the performance of these models with and without incorporating PCA for dimensionality reduction. Our hypothesis is that the model using both deep metric learning and PCA will outperform human experts in classifying VL-infected samples in image datasets.

Third, the study will evaluate the effectiveness of the classifier using established metrics sensitivity, and specificity. This evaluation will determine the classifier's ability to distinguish between positive (VL-infected) and negative (non-infected) cases with high accuracy.

In addition to focusing on VL, we have expanded our research to include the diagnosis and classification of other parasitic infections. Leveraging the best-performing models and techniques from the VL study, we will apply the most effective pipeline to classify images containing parasites from other neglected tropical diseases. By doing so, we aim to generalize the diagnostic capabilities of our approach, potentially enabling healthcare professionals to detect a broader range of parasitic diseases with the same high level of accuracy.

This research has the potential to significantly contribute to the field of medical AI. The findings could assist healthcare professionals in detecting neglected tropical diseases, leading to faster, more reliable, and cost-effective diagnoses for a variety of parasitic infections.

1.3 Related Works

Several research works have focused on leveraging machine learning and image processing techniques for detecting and classifying different types of parasites, (53) proposes a two-step method for detecting malaria parasites in thick blood smears using smartphones. The method involves using an intensity-based screening technique followed by classification with a modified Convolutional Neural Network (CNN). (17) focuses on developing accurate and efficient models for detecting parasites in single cells using various techniques. The simplified version of these models can even be utilized on mobile phones and online applications. (45) presented an automated approach for detecting Trypanosoma cruzi parasites in digital microscope images of blood smears. The approach combines image pre-processing steps like mask generation and filtering with a KNN classifier for identifying parasites in segmented regions. Leishmania Classification Works: As recommended by WHO, the gold standard for diagnosing Visceral

Leishmaniasis (VL) in humans involves analyzing images from bone marrow parasitological examinations. Several studies have explored using machine learning for automated analysis of these images:

1.3.1 Segmentation

(15) utilized morphological and level set approaches to segment Leishmania bodies in digital microscopic images. (42) proposes a semi-automatic approach for segmenting different forms of Leishmania parasites using image processing techniques like smoothing filters and region-growing algorithms. (23) explored employing various image processing techniques like filters and gradient operators for parasite detection . (9) utilized morphological operators for segmenting Leishmania parasites.

1.3.2 Deep Learning-based Segmentation and Classification

(21) research trains a U-Net (40) model to segment and classify Leishmania parasites into different forms like promastigotes, amastigotes, and adhered parasites. (19) employed a U-Net architecture to automatically identify pixels containing Leishmania parasites. The training process is guided by expert annotations to achieve better accuracy.

The experiments of (15; 42; 23) were performed over a public dataset provided by (15) whereas (40; 21; 19) conducted their experiments in non-public datasets.

2

Theoretical Background

2.1 Digital Image Processing and Computer Vision

To understand what is Digital Image Processing (D.I.P) first we need to understand what is and digital image. An image is characterized as a two-dimensional function, denoted as f(x, y), where x and y represent spatial coordinates, and the amplitude at any coordinate pair signifies the image intensity or gray level. When x, y, and intensity are finite and discrete, the image is termed a digital image (18). Digital image processing involves the manipulation of digital images using a computer. The images consist of finite elements known as pixels or picture elements, with each pixel having a specific location and value. Vision, being a primary human sense, relies heavily on images; however, machines can process images from a wide electromagnetic spectrum, including sources like ultrasound and electron microscopy.

The use of digital image processing can be traced back to the early 1960s when computers with sufficient power for meaningful image processing tasks emerged. The pioneering application of computer techniques to enhance images taken by space probe's from the moon and simultaneously, in the late 1960s and early 1970s, digital image processing techniques began finding applications beyond space exploration. In medical imaging, remote Earth resources observations, and astronomy, these techniques started making significant contributions. A landmark development during this period was the invention of computerized axial tomography (CAT) in the early 1970s, a crucial advancement in medical diagnosis. CAT involves a ring of detectors encircling an object, rotating around it while an X-ray source collects data. Algorithms then reconstruct a 3D image, allowing for detailed medical imaging. Sir Godfrey N. Hounsfield and Professor Allan M. Cormack, the inventors of tomography, shared the 1979 Nobel Prize in Medicine.

The application of digital image processing continued to evolve from the 1960s to the present, experiencing substantial growth. Beyond medicine and space exploration, these techniques found utility in diverse fields. In medical contexts, computer procedures enhance contrast

Figura 1: Aortic Angiogram. Source: (18).

or code intensity levels into color for improved interpretation of X-rays and other images. Geographers deploy similar techniques to analyze pollution patterns using aerial and satellite imagery. Image enhancement and restoration procedures are crucial in processing degraded images of unrecoverable objects or expensive-to-duplicate experimental results. Angiography stands as another major application in an area called contrast enhancement radiography, primarily employed for generating images of blood vessels known as angiograms. This intricate procedure involves the insertion of a catheter, a small, flexible, hollow tube, typically introduced into an artery or vein in the groin region. The catheter is carefully maneuvered through the blood vessel, directed towards the specific area under scrutiny. Once the catheter reaches the targeted site, an X-ray contrast medium is injected through the tube. This infusion of contrast medium serves to heighten the visibility of blood vessels, facilitating radiologists in identifying any irregularities or blockages within the circulatory system

Computer vision has evolved into a broad and expansive field, encompassing the recording of raw data and progressing towards the extraction of image patterns and information interpretation. Rooted in a combination of concepts from digital image processing, pattern recognition, artificial intelligence, and computer graphics, computer vision engages in diverse tasks related to obtaining information from input scenes, primarily digital images, and subsequent feature extraction. The methods employed in addressing problems within computer vision are contingent on the specific application domain and the characteristics of the data being analyzed.

Within the interdisciplinary realm, computer vision is recognized as a fusion of image processing and pattern recognition. The culmination of the computer vision process is image un-

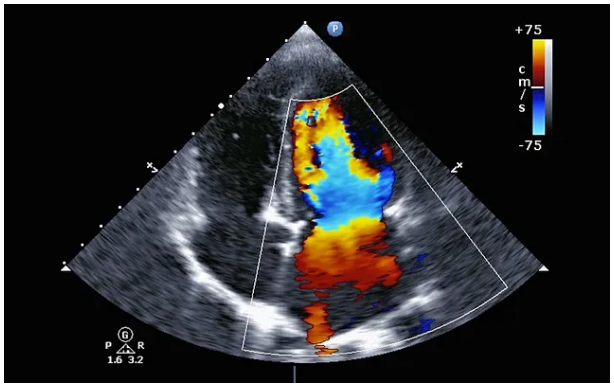

Figura 2: Echocardiogram. Source: (hvc).

derstanding, a development that borrows from the adaptive abilities of human vision in acquiring information. Computer Vision serves as the discipline focused on extracting information from images, distinct from the objectives of Computer Graphics. Its advancement is intricately tied to the evolution of computer technology systems, influencing aspects like image quality improvement and image recognition. While there exists an overlap with Image Processing in basic techniques, some authors use these terms interchangeably.

The primary goal of Computer Vision is to create models, extract data, and derive information from images. In contrast, Image Processing revolves around implementing computational transformations for images, including actions like sharpening and contrast adjustments.

Over the recent years, there has been a remarkable surge in the number of publications utilizing computer vision techniques for the analysis of static medical imagery, escalating from hundreds to thousands. This proliferation of research activity is particularly pronounced in certain medical specialties, notably radiology, pathology, ophthalmology, and dermatology. The heightened interest in these areas can be attributed to the visual pattern-recognition nature inherent in diagnostic tasks within these specialties. Additionally, the increasing accessibility of highly structured medical images has fueled the momentum in applying computer vision methodologies to enhance diagnostic capabilities in these fields (13).

Cardiac imaging, particularly echocardiography, has witnessed substantial advancements through deep learning. The cost-effective and radiation-free nature of echocardiograms makes them suitable for AI applications. Deep learning models, have demonstrated the ability to recognize cardiac structures, estimate function, and predict systemic phenotypes..

In dermatology, deep learning has excelled in lesion-specific diagnostics, demonstrating capabilities comparable to board-certified dermatologists. Convolutional Neural Networks (CNNs) have been successful in classifying malignant skin lesions and, more recently, have expanded to differential diagnostics across various skin conditions. AI algorithms hold promise in supporting large-scale detection of malignancies and tracking lesion growth over time.

2.2 Classification Models

The classification task involves a computer program determining the categorical affiliation of a given input among k predefined categories. Typically, the learning algorithm generates a function f:Rn→ 1, . . . , k, where y=f(x) assigns a numeric code to an input vector x, signifying its classification. Variations of this task include scenarios where f outputs a probability distribution over classes.

Illustrating a classic example of classification, object recognition in images entails assigning a numeric code to identify the object depicted, with the input represented as pixel brightness values. The complexity of classification is heightened when the computer program cannot guarantee the availability of every measurement in its input vector, presenting a common challenge in medical diagnosis where certain tests are costly or invasive.

Addressing this challenge involves adapting the learning algorithm to contend with missing data. Rather than formulating a distinct classification function for each potential set of missing inputs, we propose a novel probabilistic framework. By learning a probability distribution over all relevant variables, the classification task is effectively solved by marginalizing out the missing variables.

This approach streamlines the definition of an extensive set of classification functions, each dedicated to classifying x with a unique subset of missing inputs. Particularly pertinent in medical diagnosis, this methodology significantly reduces the computational burden, as the program only needs to learn a single function encapsulating the joint probability distribution of all input variables. We discuss the application of this approach in challenging classification tasks, demonstrating its effectiveness in scenarios where certain inputs may be absent.

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) (25) have garnered significant attention in recent years for their remarkable performance in various image analysis tasks, particularly in the ImageNet Large Scale Visual Recognition Challenge (41). This challenge serves as a benchmark for evaluating state-of-the-art image classification and segmentation algorithms. The success of CNN-based deep neural networks in this challenge underscores their potential in the field of medical image classification, where accurate and efficient classification is crucial for diagnosis and treatment. CNNs, also referred to as convolutional networks, are a specialized type of neural network designed for processing data with grid-like topology. The term "convolutional"in CNNs refers to the mathematical operation of convolution, which is utilized extensively within the network architecture (20).Convolutional operations enable CNNs to automatically learn and extract features from input data, making them highly effective in tasks such as image classification and segmentation.

CNNs, in its most general form, exhibit a structured architecture composed of three distinct

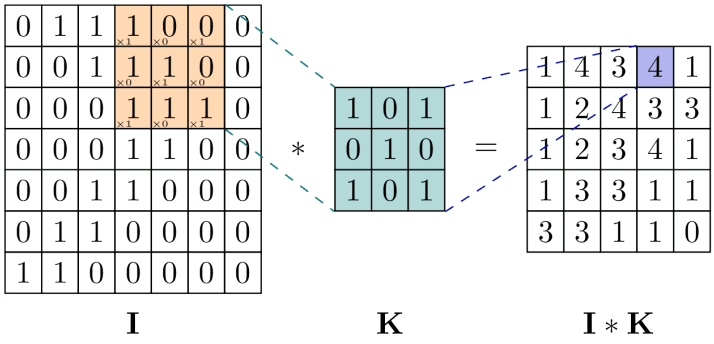

Figura 3: Convolution Example. Source: (jan).

types of layers: convolutional layers, pooling layers, and fully-connected layers. The combination of these layers forms the overall CNN architecture. The functional breakdown of the exemplary CNN can be delineated into four key areas (35):

- Input Layer: As observed in other types of Artificial Neural Networks (ANNs), the input layer serves to encapsulate the pixel values representing the image.
- Convolutional Layer: The convolutional layer is responsible for determining the output of neurons, each connected to local regions of the input. This is achieved through the computation of the scalar product between the weights associated with these neurons and the region linked to the input volume. The rectified linear unit (ReLU) activation function is commonly applied elementwise to the output, enhancing non-linearity in the model.
- Pooling Layer: Subsequently, the pooling layer performs downsampling along the spatial dimension of the input, effectively reducing the number of parameters in the activation. This process aids in capturing essential features while mitigating computational complexity.
- Fully-Connected Layers: The fully-connected layers undertake tasks analogous to standard ANNs, striving to generate class scores from the activations. It is recommended to employ ReLU activation between these layers to enhance overall performance. The produced class scores are subsequently utilized for classification purposes..

Through this sequential transformation process, CNNs systematically modify the original input layer using convolutional operations and downsampling techniques. This hierarchical approach enables the extraction of hierarchical features, leading to the generation of class scores suitable for classification and regression tasks.

2.2.2 Deep Metric Learning

To comprehend Deep Metric Learning (DML), it is crucial to first grasp the concept of metric learning. Metric learning is an approach centered on defining a distance metric directly, with

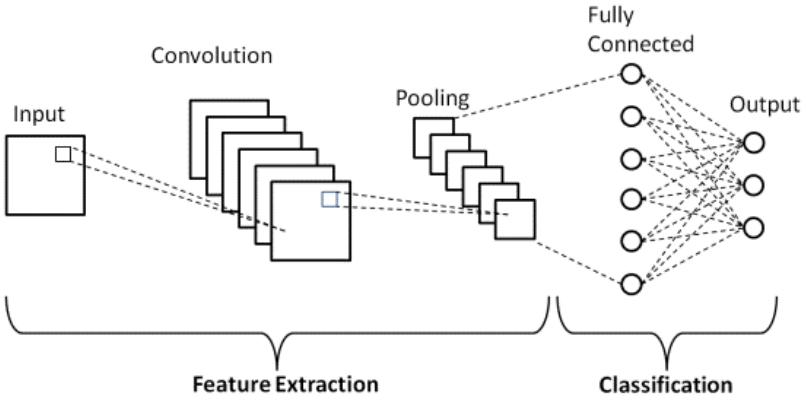

Figura 4: CNN layers. Source: (upg).

the objective of establishing the similarity or dissimilarity between objects. The fundamental goal of metric learning is twofold: to diminish the distance between objects deemed similar and concurrently augment the distance between objects considered dissimilar. This approach enables the creation of a discriminative distance metric, contributing to the enhancement of various machine learning tasks such as classification and clustering.

In recent times, the exponential growth in data volume has presented considerable benefits for achieving more precise classification results. Nevertheless, this surge in data also entails a substantial increase in computational complexity. To effectively manage the computational demands imposed by large-scale datasets, it becomes imperative to execute operations in a segmented and concurrent manner. The evolution of technology has catalyzed a significant breakthrough in Deep Learning, largely attributable to the utilization of graphics processing units, or GPUs. GPUs demonstrate exceptional proficiency in rapidly performing matrix and vector multiplications, a crucial requirement not only for creating immersive virtual realities but also for training ANN's. The adoption of GPU-based implementations in Neural Networks has made substantial contributions to recent triumphs in competitions focusing on pattern recognition, image segmentation, and object detection. The parallel processing architecture of GPUs enables the concurrent execution of numerous calculations, making them well-suited for the inherently parallelizable nature of neural network computations. Consequently, the integration of GPUs into deep learning frameworks has played a pivotal role in enhancing the efficiency and performance of neural network training, thereby contributing significantly to advancements in various domains such as pattern recognition and computer vision.(43)

The majority of current deep learning methodologies primarily rely on intricate architectural designs rather than emphasizing the development of a novel representation space through distance metrics. Nevertheless, distance-based approaches have emerged as a compelling and increasingly captivating area of focus within the realm of deep learning in recent times. (10)

In this context, Deep Metric Learning emerges as soluytion that leverages deep architectures to derive embedded feature similarity through nonlinear subspace learning. It formulates

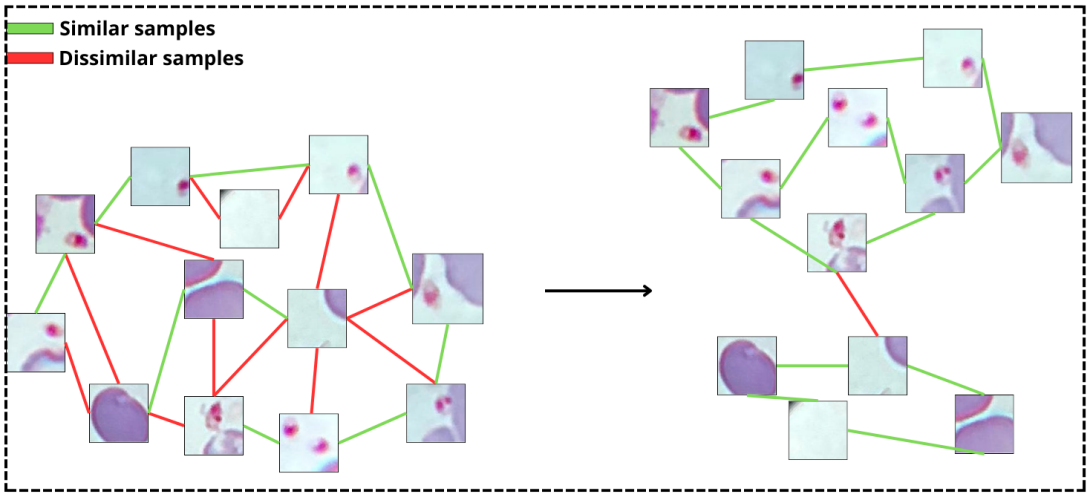

Figura 5: Metric Learning example.

problem-specific solutions by learning from raw data, addressing challenges associated with tasks such as classification, detection, segmentation, and registration in the domain of medical images.

The rapid ascent of deep learning has positioned it as a prominent and effective approach for tackling issues in medical image analysis. Notably, various problems in this domain, including classification, detection, segmentation, and registration (29), can be addressed using deep metric learning algorithms grounded in the similarity approach. By employing a deep metric, a more elevated representation level of data can be attained, facilitating enhanced analysis of medical images.

Numerous suggestions for loss functions abound in the realm of deep metric learning, each tailored to specific requirements. Examples include contrastive loss (22), triplet loss (44), quadruple loss (34), and n-pair loss (46). The careful selection of an appropriate loss function is pivotal, ensuring not only rapid convergence but also optimizing the search for the global minimum during training. This emphasis on the loss function contributes significantly to the effectiveness and efficiency of deep metric learning algorithms.

The upcoming sections will present and elucidate various types of Deep Metric Learning loss functions that will be employed in the experimental phase of this study.

Triplet

First demonstrated by (44), Triplet loss became a popular loss function used in deep metric learning, particularly for tasks like face recognition and image similarity. The fundamental idea behind triplet loss is to train a neural network to learn a feature space in which the embeddings of similar examples are closer to each other, while those of dissimilar examples are pushed apart.

The loss is formulated based on triplets of samples: an anchor (Xa), a positive (Xp), and a

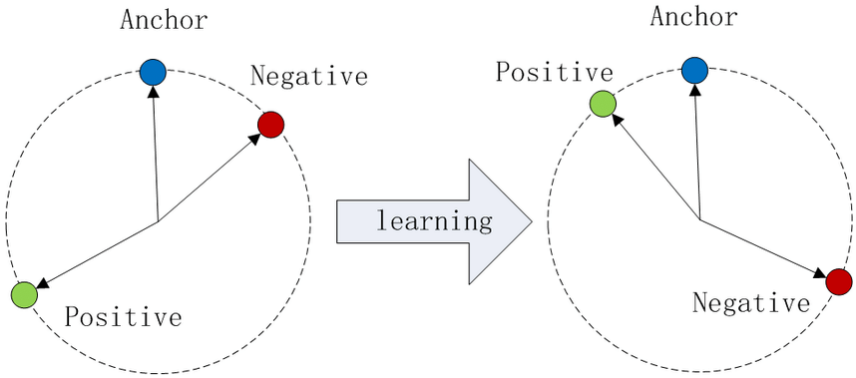

Figura 6: Triplet loss example Source: (26).

negative (Xn). These triplets are chosen from the training dataset, where the anchor and positive samples belong to the same class (or share some similarity), while the negative sample is from a different class or is dissimilar to the anchor.

The objective of triplet loss can be expressed mathematically as follows:

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

Where f(x) represents the embedding of input x, in a feature space, N is the number of triplets in the training set and α is a margin, ensuring that the positive pair (anchor and positive) is closes by at least this margin than the negative pair (anchor and negative).

Circle

(48) proposed a new novel loss function for deep metric learning particularly in scenarios where the data distribution is complex and imbalanced. Circle Loss aims to enhance the learning of deep features by adapting a re-weighting strategy for each similarity score under supervision, providing flexibility in optimization and a well-defined convergence target.

Circle loss incorporates a re-weighting mechanism for each similarity score, offering flexibility in the optimization process. This adaptability allows the loss function to better adapt to the characteristics of the data, contributing to improved learning outcomes.

The Circle loss function is designed to be compatible with both class-level labels and pair-wise labels. This adaptability enables Circle loss to seamlessly integrate into various learning scenarios. Notably, with slight modifications, Circle loss can degenerate into well-known loss functions such as triplet loss or softmax cross-entropy loss (48).

The decision boundary in Circle Loss is defined as circular in the similarity pair space, leading to a simplified and definite convergence target. This contrasts with other loss functions, where the decision boundary is linear, resulting in a more ambiguous convergence status where any point along the linear boundary is considered acceptable.

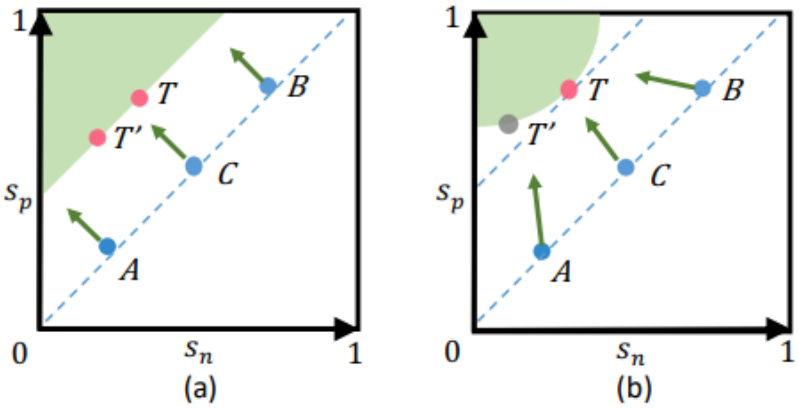

Figura 7: Circle loss example. In this example, (a) represents a popular optimization manner of reducing (sn-sp), where sn stands as the within-class similarity score and sn stands as between class similarity score, and (b) the proposed optimization using circle loss. In (a) both T and T' have the same margin, however, using Circle Loss, T would be choose to create a definite target for convergence. Source: (48).

Mathematically, Circle Loss is expressed as follows:

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

- α_{jn} and α_{ip} are non-negative weighting factors
- 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
- γ is a scale factor that controls the strength of penalization.

MultiSimilarity

Training deep metric learning models requires selecting informative pairs of data points, a crucial yet challenging task. The Multi-Similarity (MS) (50) loss tackles this challenge by harnessing three distinct types of similarities: Self-similarity, Positive relative similarity and Negative relative similarity. Self-similarity measures how similar the elements within a pair are to each other, positive relative similarity compares the similarity of a pair to other positive pairs, highlighting how distinct it is from other similar examples and Negative relative similarity compares the similarity of a pair to other negative pairs, emphasizing how different it is from dissimilar examples. By combining these similarities, the MS loss achieves an optimized pair selection and more precise weighting. Positive pairs are encouraged to have high self-similarity

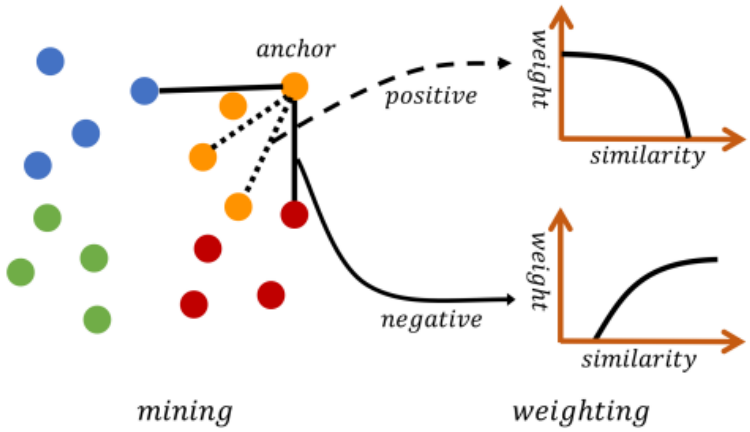

Figura 8: The multi-similarity loss is able to jointly measure the self-similarity and relative similarities of a pair, which allows it collect informative pairs by implementing iterative pair mining and weighting. Source: (50).

and positive relative similarity, emphasizing uniqueness among similar examples, while negative pairs are driven towards low self-similarity and negative relative similarity, accentuating the difference between dissimilar examples. The MS loss doesn't treat all pairs equally. Instead, it uses an iterative process through mining and weighting. Pairs are initially selected based on their positive relative similarity, focusing on unique positive example. These pairs are then refined by considering both their self-similarity and negative relative similarity, assigning higher weights to more informative pairs. The MS loss is formulated as follows:

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

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

NPairs

The N-pair loss function, introduced by (46), offers an improvement over the classic triplet loss for metric learning. While the triplet loss compares a positive example to only one negative example, the N-pair loss simultaneously compares it to multiple negative examples (N-1).

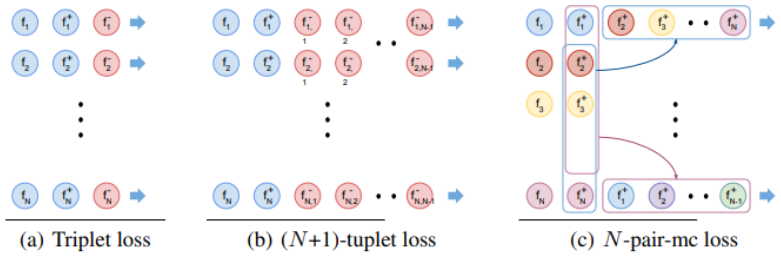

Figura 9: Triplet loss, (N+1)-tuplet loss, and multi-class N-pair loss with training batch construction. Assuming each pair belongs to a different class, the N-pair batch construction in (c) leverages all 2 × N embedding vectors to build N distinct (N+1)-tuplets with fi N i=1 as their queries; thereafter, we congregate these N distinct tuplets to form the N-pair-mc loss. For a batch consisting of N distinct queries, triplet loss requires 3N passes to evaluate the necessary embedding vectors, (N+1)-tuplet loss requires (N+1)N passes and our N-pair-mc loss only requires 2N. Source: (46).

This approach addresses a limitation of the triplet loss by considering the diversity of negative classes, leading to a more stable and balanced metric learning process.

The multi-class N-pair loss, also known as N-pair-mc loss, specifically targets multi-class scenarios. It aims to optimize the identification of a positive example among multiple negative examples from different classes. By incorporating this richer negative context, the N-pair-mc loss helps the model learn more discriminative representations. T he N-pair-mc loss can be mathematically expressed as:

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

- 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.
- N indicates the number of distinct classes represented in a batch.

2.3 Principal Component Analysis

Principal Component Analysis (PCA) (16) is a sophisticated technique employed across various fields, including data science, machine learning, and image processing. It excels in dimensionality reduction, transforming high-dimensional datasets into lower-dimensional representations while preserving the most critical information. PCA operates on the principle that the data's informative rank, or the number of essential dimensions, is typically lower than the number of original features (5). The primary objectives of PCA are to:

1. Extract the most salient information from the data table
2. Compress the dataset by retaining only this crucial information
3. Simplify the dataset description
4. Analyze the structure of both observations and variables

PCA identifies a set of new axes, termed principal components (PCs), which elucidate the majority of the variance in the data. These PCs represent new directions in the lower-dimensional space, selected to capture maximum information from the original data. The data points are subsequently projected onto these new principal components, yielding a lower-dimensional representation while preserving the most significant information (5). In this study, PCA was selected due to its capability to unveil underlying structures in complex data and mitigate noise influence. By representing the original variables with weighted averages through components, it captures essential patterns while filtering out irrelevant fluctuations. This noise minimization contributes to a clearer signal, resulting in a more robust representation (8). At its core, PCA is a mathematical technique that transforms data into a new coordinate system, prioritizing the capture of the most significant variations. The process involves several steps:

1. Calculation of the covariance or correlation matrix, summarizing relationships between all variables.
2. Determination of eigenvalues and eigenvectors of the covariance/correlation matrix. Eigenvalues represent the variance captured by each principal component, while eigenvectors define the directions in the new coordinate system.
3. Creation of new variables (principal components) based on the eigenvectors.

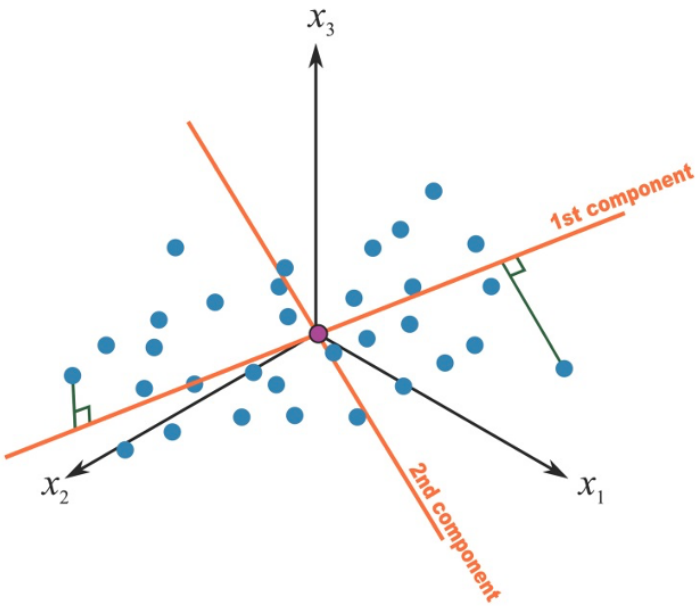

Figura 10: First and Second principal component in a 3-covariates setting. Source: (And).

The principal components are expressed as linear combinations of the original variables:

$$P_i = b_{i1}X_1 + b_{i2}X_2 + \cdots + b_{ik}X_k \tag{2.1}$$

Where P_i is the ith PC, b_{ik} is the weight for the variable X_k. It is often advantageous to standardize all variables X_k to zero mean and unit standard deviation. Each PC is a linear combination of the original variables, with weights determining the contribution of each original variable to the corresponding principal component. PCA computes these new variables (principal components) under specific constraints. The first principal component is required to have the largest possible variance, thus explaining the largest part of the data's inertia. Subsequent components are computed to be orthogonal to the previous ones while maximizing the remaining variance (5). The components in PCA are obtained from the Singular Value Decomposition (SVD) of the data table $\mathbf{X}$. If $\mathbf{X} = \mathbf{P}\blacksquare\mathbf{Q}^T$, the $I \times L$ matrix of factor scores, denoted $\mathbf{F}$, is obtained as:

$$\mathbf{F} = \mathbf{P}\blacksquare \tag{2.2}$$

The matrix $\mathbf{Q}$ provides the coefficients of the linear combinations used to compute the factor scores. It can also be interpreted as a projection matrix, as multiplying $\mathbf{X}$ by $\mathbf{Q}$ gives the projections of the observations on the principal components:

$$\mathbf{F} = \mathbf{P}\blacksquare = \mathbf{P}\blacksquare\mathbf{Q}^T\mathbf{Q} = \mathbf{X}\mathbf{Q} \tag{2.3}$$

Geometrically, the components can be represented as a rotation of the original axes. The factor scores give the length of the projections of the observations on the components. In this context, $\mathbf{Q}$ is interpreted as a matrix of direction cosines (due to its orthonormality) and is also referred to as a loading matrix (5). The original data matrix $\mathbf{X}$ can be reconstructed as the product of the factor score matrix and the loading matrix:

$$\mathbf{X} = \mathbf{F}\mathbf{Q}^T \text{ with } \mathbf{F}^T\mathbf{F} = \blacksquare^2 \text{ and } \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \tag{2.4}$$

This decomposition is often referred to as the bilinear decomposition of $\mathbf{X}$ (5). By leveraging these mathematical properties, PCA provides a powerful tool for data analysis, compression, and visualization across various domains in data science and machine learning.

3

Methods

3.1 Preprocessing

In image analysis, pre-processing refers to the initial steps of preparing the data before it is fed into a model for analysis. In this study, pre-processing was applied to full-sized images of biological samples known to contain Leishmania amastigotes, as well as other parasites. The same pre-processing techniques were generalized to handle additional parasites, ensuring consistency across the dataset.

In the field of micrography, achieving good contrast is crucial for accurately identifying and quantifying individual structures within an image. Without sufficient contrast, these tasks become imprecise or even impossible. To address this challenge, we employed contrast enhancement, a digital image processing technique that manipulates the intensity values of pixels within an image. This process typically involves increasing the difference in intensity between the image's brightest and darkest regions, resulting in a visually clearer and more detailed picture that facilitates analysis. This method was applied not only to Leishmania samples but also expanded to improve the clarity of images containing other parasitic organisms, ensuring that the model's ability to detect and classify diverse parasites was optimized. 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:

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

- 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 Classifiers

One of the fundamental tasks in machine learning is classification. In this task, the goal is to automatically categorize data points into pre-defined groups, known as classes, utilizing a learning algorithm.

Data points are the individual pieces of information used for classification. They can be numerical values, text, images, or any other format relevant to the task. In our study, the data points are the images from bone marrow aspirates, initially focusing on samples containing Leishmania parasites but later expanded to include images of other parasites as well.

The pre-defined groups are the categories the data points can belong to. In our study, the categories were initially "leishmania"and "not leishmania", but these categories were later extended to encompass a broader range of parasite types. The learning algorithm is the core of the classification system. It analyzes a training dataset containing labeled data points to learn the characteristics that distinguish different categories. The learning algorithm typically learns a mapping function, denoted as f(x). This function takes an input data point x and assigns it a code representing its predicted category.

In some cases, the function might output a probability distribution indicating the likelihood of the data point belonging to each potential category.

A common application of machine learning classification in healthcare is the one we are conducting in this experiment, predicting if a person is infected with a specific disease. In our experiment, an algorithm was trained on image data from patients diagnosed with Leishmania and patients not infected. This setup was later generalized to include data from other parasitic infections, allowing the model to learn the characteristics associated with multiple diseases. The trained model can then be used to analyze data from new patients and predict whether they are infected with Leishmania or another parasite, or if they are not infected at all.

3.2.1 Classic Classifiers

K-Nearest Neighbors

K-Nearest Neighbors (KNN) stands as a fundamental and widely used technique within the realm of supervised learning, particularly renowned for its classification tasks. This intuitive algorithm operates by analyzing the similarity,often measured through distance metrics, between an unseen data point and the labeled data points within a training dataset.

During the classification process, KNN identifies the k closest data points to the unseen point based on the chosen distance metric. Subsequently, it determines the most prevalent class amongst these k neighbors, assigning that class label to the unseen point. This majority vote approach underpins the core principle of KNN classification.

The value of k, which signifies the number of neighbors to consider, plays a crucial role in KNN's performance. Choosing an appropriate k is essential, as a low k can lead to overfitting,

while a high k might result in underfitting, as the model fails to capture the underlying patterns within the data.

Despite its relative simplicity, KNN exhibits remarkable interpretability, allowing for a clear understanding of the rationale behind its classifications. This transparency, coupled with its efficiency and versatility in addressing various classification problems, makes KNN a valuable tool for both novice and experienced data scientists.

To our purpose, KNN will be used to classificate the data points in the new feature space embedding created by the DMLs.

Support Vector Machines

Support Vector Machines (SVMs) constitute a prominent supervised learning paradigm frequently employed in both classification and regression tasks. Their distinguishing characteristic is their inherent capability to identify the optimal hyperplane, or ensemble of hyperplanes, within high-dimensional spaces, effectively separating distinct data classes.

This separation is achieved by strategically positioning the hyperplane to maximize the margin, which represents the distance between the hyperplane and the closest data points from each class, known as support vectors. To accomplish this, SVMs can leverage kernel functions, which essentially map the data points into a higher-dimensional space where linear separation becomes attainable. This process enhances the distinction between categories, particularly when dealing with complex, non-linear relationships often present within data sets containing numerous variables.

Consequently, SVMs demonstrate exceptional generalizability, making them well-suited for diverse data types and applications. Their robust performance in classifying and estimating data points positions SVMs as valuable tools across various scientific and practical domains.

In our experiments, we willuse SVMs as the classification model for our DML embeddings, in two scenarios: directly classifying the datapoints given by DML network, and classifying the datapoints after PCA dimensionality reduction.

PCA Classifier

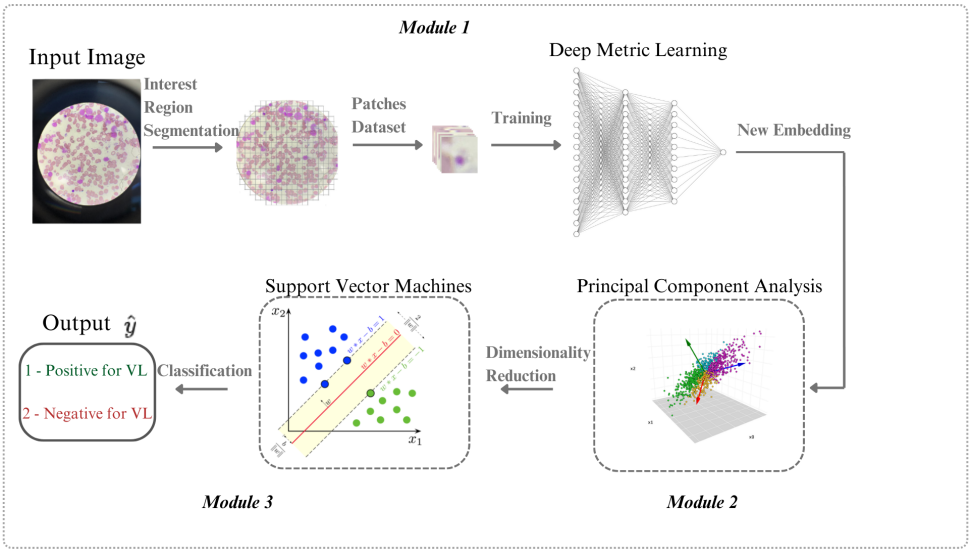

Figura 11: Proposed method for leishmania parasite classification.

Our proposed pipeline for classification utilizes, in addition to DML to create a new feature space embedding and SVM to classify the datapoints, PCA to reduce the new embeddings dimension with the purpose to reduce noise and get better results. To achieve the best possible representation of the data in the reduced space, we need to select an appropriate number of dimensions to retain. This typically involves analyzing the variance associated with each principal component. In this study, we identified the number of principal components that collectively captured 90% of the total variance in the data. This approach ensures that we retain the most information-rich dimensions while minimizing redundancy.

4

Experimental Results and Discussions

4.1 Datasets

4.1.1 Leishmania

Two separate image collections were used to create a Leishmania dataset for the experiments. The first dataset, provided by (14), contains 45 pairs of color microscope images of bone marrow aspirates. These images were captured using a Sony DSC H9 digital camera attached to an Olympus-CH40RF200 optical microscope [1].

The second dataset, created by (31), consists of 68 pairs of images. These images were captured using an iPhone 8 attached to a 1000x magnification optical microscope. Each pair includes a corresponding image, as shown in the figure, with the same dimensions as the original image. The white regions in these images highlight the locations of parasites within the original RGB image.

The figure 12 shows that the microscope's external area is present in the images of dataset 2. Since our method relies on patch analysis, keeping this periphery would create many irrelevant patches. This would slow down the algorithm and reduce its overall classification accuracy. To address this, (28) proposed using the Hough Circles algorithm. This algorithm identifies the circle encompassing the microscope. Then, a binary image is created to isolate this region. Finally, the minimum bounding box is extracted for precise segmentation of the Region of Interest (ROI). 13 shows an example of an dataset 2 image without the microscope's after using the algorithm.

Dataset 1 required additional preprocessing to handle images with parasite markings. Binary masks were generated corresponding to the RGB images, similar to the masks in Dataset 2.

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

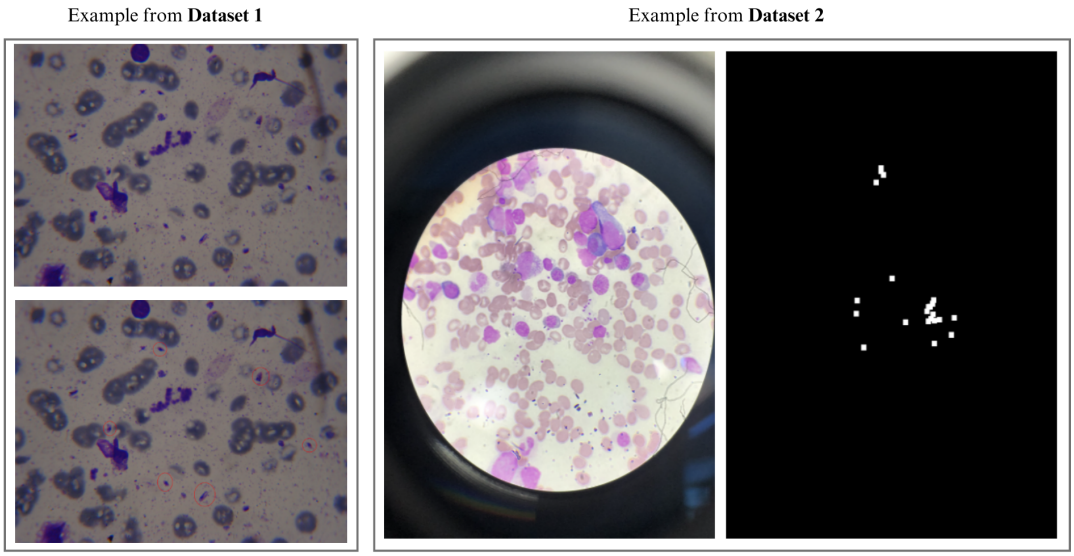

Figura 12: Examples from the two Leishmania datasets. In left we have a pair of images from dataset 1, in top we have the original image, at bottom we have the original image marked where it contains parasites. In right we have a pair of images from dataset 2, first the original image, with the microscope being visible, and second we have the mask with the location of the parasites in the orginal image.

These masks highlight the parasite locations and were created manually using `Photoshop`[2] software. 13 shows an example of generated mask from dataset 1 images.

4.1.2 Plasmodium species

MP-IDB (30) is a publicly available image dataset consisting of 229 images representing four different species of Malaria parasites: Falciparum (122 images), Malariae (37 images), Ovale (29 images), and Vivax (41 images). For each species, there are images corresponding to four distinct life stages: Ring, Trophozoite, Schizont, and Gametocyte. Expert pathologists have provided the ground-truth for each image, ensuring accurate annotations. The images are captured in 24-bit RGB color format with a microscope magnification of 100x.

4.1.3 Trypanosoma cruzi

TRYP-DB (32) is a publicly available image dataset introduced by (33). It consists of 33 slides containing thin blood smears from Swiss mice, experimentally infected with the T. cruzi Y strain during the acute phase. These slides were prepared for image annotation and analysis. The samples were obtained from animals housed at the Chagas Disease Laboratory at the Federal University of Ouro Preto, where the T. cruzi strain was maintained through successive blood

[2]Available at `https://www.adobe.com/products/photoshop.html`

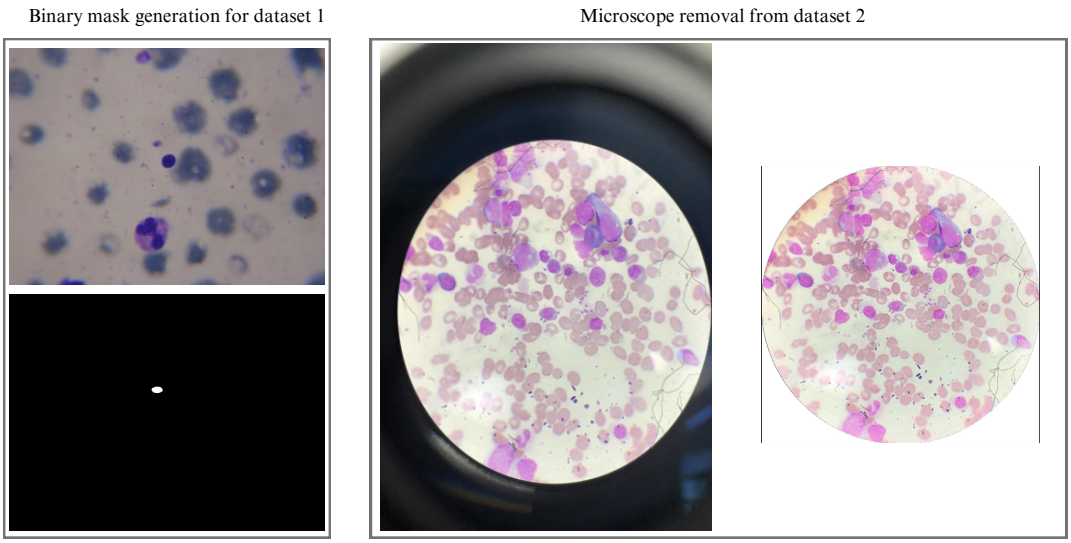

Figura 13: Examples from the two leishmania datasets after the pre process stage.

passages in mice.

4.1.4 Schistosoma

SH-DB (37) is a publicly available dataset developed by (36). It contains 12,051 microscopic images derived from 103 urine samples, along with their corresponding segmentation masks, which were manually annotated for Schistosoma haematobium eggs. The samples were collected from school-age children who had reported the presence of blood in their urine. A standard urine filtration procedure was used to process the clinical samples, where 10 mL of urine was passed through a 13 mm diameter filter membrane with a pore size of 0.2 μm. After filtration, the membrane was placed on a microscope slide and covered with a coverslip to ensure flatness for image capture. The images were obtained using a digital microscope called the Schistoscope.

4.1.5 All: Plasmodium, Toxoplasma, Babesia, Leishmania, Trypanosome and Trichomonad

ALL-DB (27) is a publicly available dataset consisting of 34298 microscopic images of multiple parasites (Plasmodium, Toxoplasma, Babesia, Leishmania, Trypanosome, Trichomonad) and host cells (Red blood cells and Leukocytes) under 400X or 1000X magnification. Specifically, the dataset includes 843 images of Plasmodium and 3758 images of T. gondii at 400X, and 2933 images of Toxoplasma, 1173 images of Babesia, 2701 images of Leishmania, 2385 images of Trypanosome, and 10134 images of Trichomonad at 1000X. Additionally, it contains

8995 images of Red blood cells and 461 images of Leukocytes at 1000X, along with 915 images of Leukocytes captured at 400X.

4.2 Data Augmentation

In this study, for the leishmania dataset, we generated synthetic data for the positive class by rotating images up to 120 degrees, flipping them horizontally and vertically, and zooming in slightly (up to 10%). These variations increased the dataset's diversity. Machine learning research shows that data augmentation can improve model performance, especially for limited datasets. This technique creates more training data by applying geometric transformations to existing samples. However, it's crucial to choose appropriate transformations and hyperparameters to avoid distortions.

It's important to remember that augmented data isn't entirely new information. It's essentially modified versions of existing data. Overusing it can lead the model to focus on features specific to these synthetic samples, hindering its ability to generalize to real-world data.

To prevent overfitting, we downsampled the majority class (negative class) after augmentation. Downsampling randomly removes a specific number k of samples from the majority class. In our case, k was chosen to achieve a final dataset ratio of positive to negative samples of 1:2, $k = N - 2P$, where N is the size of the negative class and P is the size of the positive class.

4.3 Data Sampling

For the Leishmania datasets, following the RGB image augmentation described in section 4.2, we obtained corresponding binary masks for each image. We then merged these two datasets, images and masks, before applying further data balancing techniques.

To address the class imbalance, we performed data augmentation for the positive class, the minority class, and downsampling for the negative class, the majority class. This process resulted in a final dataset of 65,202 balanced images (each 96x96x3 pixels in size). The details of the data split for training, validation, and testing are presented in Table 1.

To achieve optimal performance, deep learning models benefit from being trained on well-structured datasets. This typically involves splitting the data into three subsets: training, validation, and testing.

The training set is the workhorse of the process. It provides the model with labeled examples, allowing it to learn and refine its internal parameters. The validation set acts as a guide during training. By monitoring the model's performance on this set, we can adjust hyperparameters and prevent overfitting. The testing set offers an unbiased evaluation of the final model's generalizability. It consists of unseen data, and the model's performance on this set reflects its real-world effectiveness. In this experiment, we adopted a common data split strategy, alloca-

	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 1: Quantity of data through data wrangling stages.

ting 70% of the data for training, 15% for validation, and 15% for testing. The specific details of this division are presented in Table 2.

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

Tabela 2: Quantity of patches for training, validation and, testing.

As an extended part of our study, we analyzed additional datasets beyond the core investigation into Leishmania classification. These datasets—MP-IDB, TRYP-DB, SCH-DB, and ALL-DB were explored without any data augmentation or downsampling techniques applied. The following tables present the quantity and class distribution of images used from these datasets, detailing both binary classification tasks (positive-negative) and multiclass classification tasks.

	Original size	**Patches (positive-negative)**	**Synthetic Data**	**Randomly removed from negative class**
MP-IDB Malaria	37	397 - 19836	-	-
MP-IDB Vivax	41	1039 - 20095	-	-
MP-IDB Ovale	29	476 - 15482	-	-
MP-IDB Falciparum	122	7251 - 58970	-	-
TRYP-DB	674	1869 - 15747	-	-
SCH-DB	12051	7790 - 37765	-	-

Tabela 3: Quantity of data for MP-IDB, TRYP-DB, and SCH-DB datasets.

Class Label	Class Name	Number of Images	Magnification
0	Babesia	1173	1000X
1	Leishmania	2701	1000X
2	Leukocyte	915	400X
3	Plasmodium	843	400X
4	Toxoplasma	2933	1000X
5	Trypanosome	2385	1000X

Tabela 4: Class distribution and magnification levels in the ALL-DB dataset.

4.4 Hyperparameters and models architectures

4.4.1 Metric Losses

In this work, the `Python Metric Learning`[3] was used to run and perform experiments with different metric learning losses. This packages provides access to various DML loss functions and can be incorporated easily to any deep learning pipelines and architecture. The four losses we experimented with and its parameters will be explained bellow.

Triplet Loss

This loss employs two key parameters, margin and distance metric. A margin of 0.3 was set, it ensures positive examples are closer to a reference point than negatives, creating a clear decision boundary. Additionally, cosine similarity, the chosen distance metric, prioritizes the angle between data points, emphasizing their shape over size. This approach is crucial as parasite structures are more reliable identifiers than size for classification.

Circle Loss

Two additional parameters are crucial for this loss decision-making process: Relaxation factor and gamma. Relaxation Factor (m) controls the radius or flexibility of the decision boundary. The model sets m to 0.4, following the precedent set in prior research (48). In that work, the authors used 0.25 for face recognition tasks and 0.4 for fine-grained image retrieval. The chosen value of 0.4 likely reflects a balance between accuracy and robustness, considering the model's application. The Gamma Parameter (γ) plays a role in shaping the decision boundary. The model adopts a value of 256 for gamma, again aligning with the findings in (48). There, the authors used 256 for face recognition and 80 for fine-grained image retrieval. The higher value of 256 might be necessary due to the potentially greater variability in parasite shapes compared to human faces.

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

Multisimilarity Loss

This loss relies on three key hyper-parameters: Alpha (α), Beta (β) and the Margin (λ). Alpha and Beta controls the influence of different types of training pairs on the model's learning process. The chosen configuration ($\alpha = 2$ and $\beta = 50$) emphasizes the contribution of informative pairs, likely containing parasites, by assigning a higher weight (β) to their loss function compared to less informative pairs (α). This helps the model prioritize learning from valuable examples. The margin sets a similarity threshold for training pairs. Pairs exceeding this threshold in similarity, likely positive examples, are considered similar, while those falling below it , likely negative examples, are deemed dissimilar. The chosen value ($\lambda = 1$) establishes the difficulty of the learning task. A higher value would create a more relaxed threshold, making it easier for the model to identify similar pairs.

NPairs Loss

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

4.4.2 CNN architectures

VGG19

- Convolutional Layers:
  - VGG19 consists of a series of five convolutional blocks. Each block stacks three convolutional layers with these properties: Kernel size: 3x3, Stride: 1 , Padding: Same. The number of filters used in the convolutional layers increases as you progress through the network, starting from 64 and going up to 512.
- Pooling Layers:
  - Max pooling layers are inserted after each convolutional block.These layers reduce the spatial dimensionality of the feature maps by taking the maximum value within a specific window, in our case 2x2.
- Activation
  - Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation function.
- Fully Connected Layers
  - After the convolutional blocks, there are two fully-connected layers with 4096 neurons each, and the classification layer, with its neurons equals to the number of classes, in our case, 2.

DeCaf

The DeCaf (Deep Convolutional Activation Feature) is a feature extraction method that leverages the activation outputs of a pre-trained CNN model, instead of using the final classification output of the network, DeCAF leverages activations from earlier layers within the pre-trained VGG19. These earlier layers capture more general visual features like edges and textures, making them well-suited for feature extraction tasks. The pre-trained model used for our experiments was the VGG19 model.

Base CNN for metric learning

- Convolutional Layers: The model uses three consecutive 2D convolutional layers with increasing complexity:
  - Layer 1: 32 channels, kernel size 3x3, stride 1
  - Layer 2: 64 channels, kernel size 3x3, stride 1
  - Layer 3: 128 channels, kernel size 3x3, stride 1
- Normalization and Activation: Following each convolutional layer, a batch normalization layer is applied to stabilize the learning process by normalizing the layer's outputs. A Rectified Linear Unit (ReLU) activation function is used after each convolutional and batch normalization step. This introduces non-linearity, allowing the model to learn more complex patterns. Pooling and Regularization: Two max-pooling layers, each with a 2x2 window, are inserted after the second and third convolutional layers. These layers reduce the spatial dimensions of the extracted features. Dropout layers with rates of 0.25 and 0.5 (dropout1 and dropout2) are incorporated for regularization. During training, these layers randomly set a portion of neurons to zero, helping to prevent overfitting.
- Fully Connected Layers: Two fully connected layers (fc1 and fc2) are used to compress and transform the extracted features: fc1 reduces the dimensionality to 512. fc2 maps the features to the final embedding size.
- Flattening: Before feeding into the fully connected layers, the convolutional layers' output is flattened into a one-dimensional vector. This prepares the data for linear transformation in the final layers.

4.4.3 SVM

To optimize the SVM classifier performance, a Grid Search with cross-validation was employed. This technique explores a range of hyper-parameter values to find the best configuration. Here, the grid search focused on two key parameters:

- Regularization Parameter (C): This controls the trade-off between model complexity and training accuracy. Values of 0.1, 1, and 10 were evaluated.
- Kernel Coefficient (γ): This parameter influences the influence of training data points. Values of 1, 0.1, 0.01, and 0.001 were examined. The search prioritized maximizing recall macro, a metric that considers the average performance across all classes. Additionally, the search was executed in parallel to accelerate computations. This process aimed to identify the optimal SVM configuration for each CNN model trained with different loss functions. Ultimately, the goal was to achieve efficient classification of the transformed embeddings with the highest possible recall.

4.4.4 Knn

No changes were performed in the parameters. The results shown by this classifier were obtained the with `scikit-learn` [4] package's default configuration.

4.5 Classification results

We employed stratified cross-validation on the test dataset with five folds to guarantee a balanced class distribution during model evaluation. All trained models were evaluated on the same data. We used the models to make predictions on the embeddings and calculated metrics precision, recall, F1-score, accuracy for each class. The tables present the average and standard deviation of these metrics across all folds, providing a comprehensive overview of model performance.

4.5.1 Classification with VGG19 and DeCaf

	Precision	**Recall**	**F1-Score**	**Specificity**
VGG19	0.3750 (0.0000)	0.5000 (0.0000)	0.4250 (0.0000)	0.9900 (0.0000)
DeCaf	0.3750 (0.0000)	0.5000 (0.0000)	0.4250 (0.0000)	0.9900 (0.0000)

Tabela 5: VGG19 and DeCaf - classification metrics report comparison (Positive class metrics with Specificity from Negative class)

At the start of our experiments, we tested how classical CNN models would perform on the task of Leishmania parasite classification. As can be seen from the metrics in Table 5, classical models that use multiple convolutional network layers, such as VGG19, and feature extraction models, such as DeCaf, encounter significant difficulty in correctly classifying the positive class.

[4] Available at https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier

It can be noted that they are effective in correctly classifying the negative class, which can be explained by the larger quantity and variety of data used in training. The limitations of such models are also evident from the standard deviation results, which indicate low variability in their performance.

4.5.2 Classification with DML + knn

	Precision	**Recall**	**F1-Score**	**Specificity**
Triplet	0.3566 (0.0321)	0.7633 (0.0115)	0.4866 (0.0321)	0.5266 (0.0750)
Circle	0.8900 (0.0100)	0.7400 (0.0100)	0.8066 (0.0115)	0.9666 (0.0057)
Multisimilarity	0.5333 (0.2746)	0.7800 (0.0264)	0.6100 (0.1652)	0.6866 (0.2294)
NPairs	0.6666 (0.0057)	0.4366 (0.0115)	0.5266 (0.1154)	0.9266 (0.0057)

Tabela 6: KNN - Classification metrics report comparison (Positive class metrics with Specificity from Negative class)

Our first approach to classifying embeddings generated by the networks with metric loss was to classify them with the classic distance-based classification algorithm, KNN. As can be seen in Table 6, the results obtained in this way showed a good evolution compared to the CNN models initially tested. In this scenario, it is possible to identify the Circle Loss and Multisimilarity with higher and more balanced metrics results, with the Triplet also showing promising results in its recall and the NPairs with the lowest results. It is interesting to note the greater variation in these results, which is indicated by the standard deviation, indicating that the KNN distance decision, although robust on average in the vectors, may present some performance losses in the metrics.

4.5.3 Classification with DML + SVM

	Precision	**Recall**	**F1-Score**	**Specificity**
Triplet	0.9130 (0.0050)	0.9130 (0.0050)	0.9130 (0.0030)	0.9130 (0.0050)
Circle	0.9130 (0.0050)	0.9200 (0.0000)	0.9130 (0.0050)	0.9700 (0.0000)
Multisimilarity	0.8900 (0.0100)	0.9130 (0.0050)	0.9030 (0.0020)	0.9630 (0.0030)
NPairs	0.6160 (0.0050)	0.4760 (0.0132)	0.5260 (0.0800)	0.9000 (0.0030)

Tabela 7: SVM - Classification metrics report comparison (Positive class metrics with Specificity from Negative class)

Aiming to improve the results obtained by the initial KNN-based embedding classification method, we conducted further experiments using SVM to classify these vectors. The results

can be seen in Table 7. They show a significant performance increase in all computed metrics for positive and negative classes, with all recalls, except for NPairs which had smaller gains, reaching values greater than 90

We can again highlight Triplet, Circle, and Multisimilarity, which showed promising recall results, with Circle achieving the highest value of 92%. In this scenario, NPairs also obtained the lowest results. A decrease in the variability of the results can also be observed, as seen in the decrease in the standard deviation of the metrics. This indicates that the SVM classification method can better adapt to the distinction of images from both classes.

After this result, a model was proposed that uses PCA before SVM classification to further improve the already very good results.

4.5.4 Classification with DML + SVM + PCA

	Precision	**Recall**	**F1-Score**	**Specificity**
Triplet	0.9635 (0.0042)	0.8283 (0.0115)	0.8908 (0.0076)	0.9841 (0.0018)
Circle	0.9872 (0.0024)	0.9830 (0.0046)	0.9850 (0.0020)	0.9935 (0.0012)
Multisimilarity	0.8862 (0.0167)	0.9613 (0.0057)	0.9221 (0.0084)	0.9373 (0.0104)
NPairs	0.8941 (0.0083)	0.9504 (0.0119)	0.9213 (0.0087)	0.9430 (0.0046)

Tabela 8: PCA model - Classification metrics report comparison (Positive class metrics with Specificity from Negative class)

Based on Tables 8, the proposed model using PCA for dimensionality reduction achieves higher prediction quality than previously seen models. Specifically, the model using the Circle Loss that outperforms all others in all computed metrics. From the tables, we can infer certain behaviors about each loss and the effect that dimensionality reduction had on its performance. In the case of Triplet Loss, there was a gain in the correct prediction of different negative examples, which can be seen by its high recall. However, there was a drop in the recall of the positive class compared to the results without dimensionality reduction (7). The low recall shows that, for Triplet Loss, there was a tendency for more false negatives in the prediction of the model with PCA compared to the model without PCA.

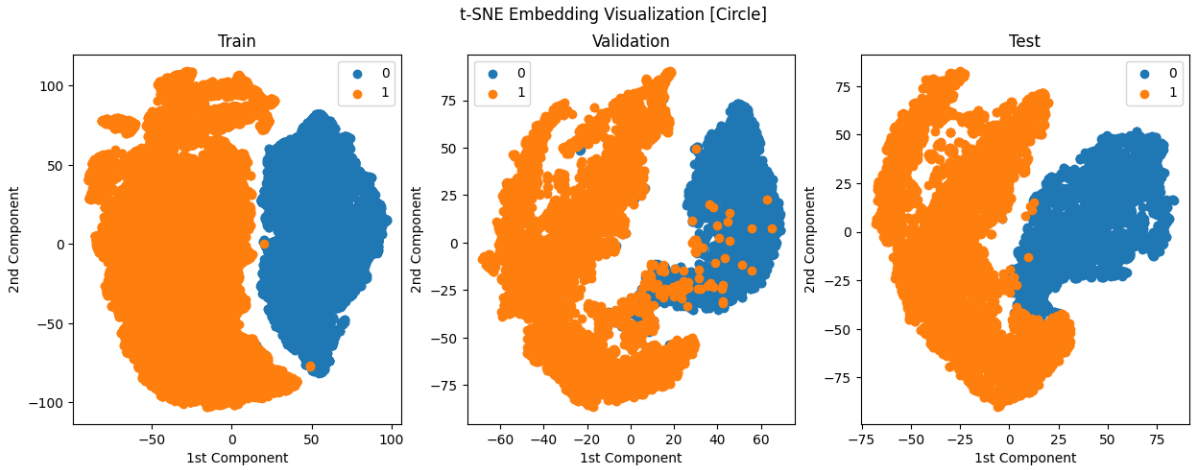

Figura 14: Visualizing new data representation (embedding space) created by Circle loss.

The Circle Loss, which already had a good result in the model without PCA, being the highest recall, became our best model after the application of PCA, indicating a good increase in correct predictions. This can be seen in both the positive and negative classes.

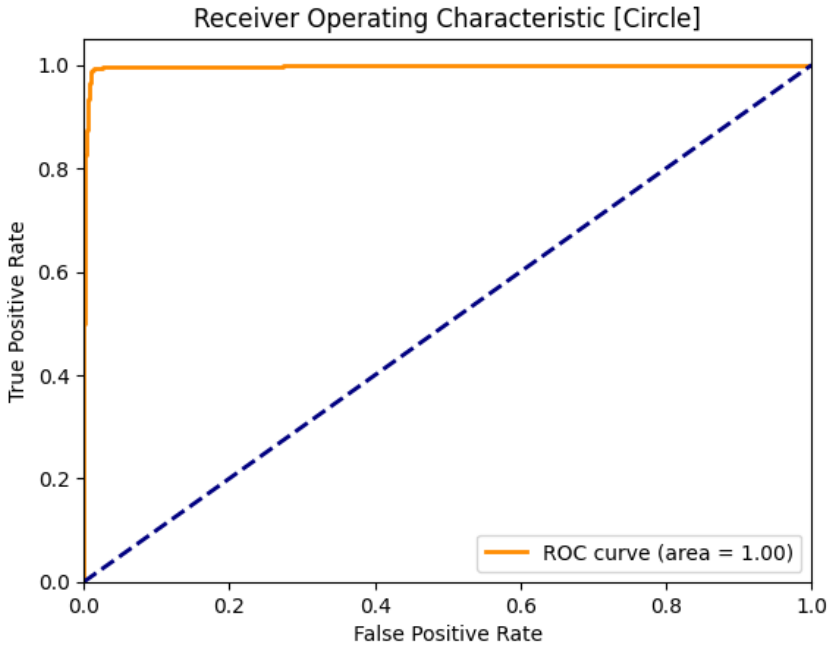

Figura 15: ROC curve from Circle model on test set.

MultiSimilarity also obtained better results after dimensionality reduction, as can be seen by its higher recall than the model without PCA, indicating a decrease in false negatives. There was a small drop in the recall of the negative class, indicating an increase in false negatives for this class.

Npairs, which in previous experiments had been the loss with the worst results compared to

others, obtained a significant improvement in its metrics when working with a reduced embedding vector. As 7 and 8 show, the recall obtained an increase of 47% in its mean, indicating a large decrease in false negatives. It can also be seen, by analyzing the standard deviation, that the npairs model is the one with the highest standard deviation in both models, indicating a greater variation in its results.

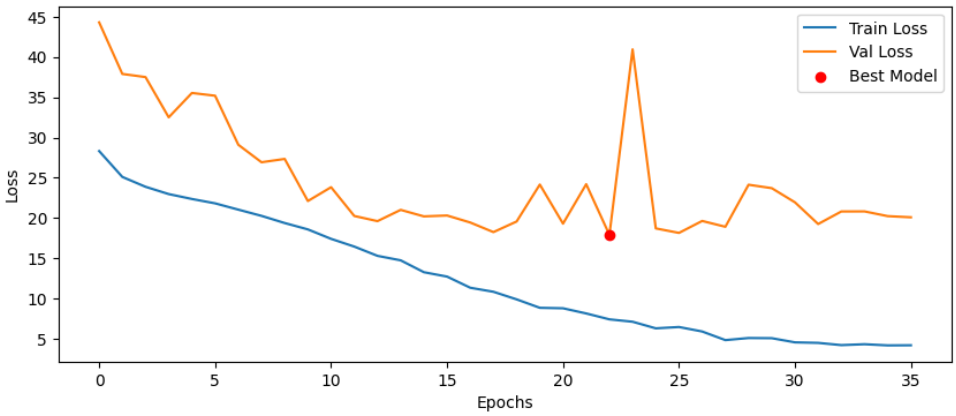

Figura 16: Circle model training history.

4.5.5 Other parasites

We extended the evaluation of the models to different parasite datasets to test their performance across a wider range of parasites. The table summarizes the classification results, showing the best-performing deep metric learning loss function for each dataset. All results were obtained using a combination of deep metric learning with SVM and PCA for dimensionality reduction.

	Loss Function	**Precision**	**Sensibility**	**F1-Score**	**Specificity**
MP-IDB (30)	Triplet	0.987 (0.002)	0.983 (0.004)	0.985 (0.002)	0.993 (0.001)
TRYP-DB (32)	Multisimilarity	0.993 (0.008)	0.943 (0.034)	0.967 (0.017)	0.999 (0.001)
SH-DB (37)	Multisimilarity	0.703 (0.019)	0.864 (0.015)	0.775 (0.015)	0.926 (0.006)

Tabela 9: Classification metrics report comparison across datasets (Positive class metrics with Specificity from Negative class)

As shown in 9, the MP-IDB dataset, using the Triplet loss function, achieved high precision (0.987) and sensitivity (0.983). For the TRYP-DB dataset, the Multisimilarity loss function also produced strong results, though the sensitivity was lower at 0.943. In contrast, the SH-DB dataset, while achieving moderate precision (0.703), had a relatively higher sensitivity (0.864). These results highlight the models' adaptability and effectiveness across varying datasets.

To further explore the capabilities of our model, we applied it to the MP-IDB dataset, focusing on diagnosing the four different species of Plasmodium parasites. The table 10 presents

the classification metrics for P. malariae, P. vivax, P. ovale, and P. falciparum using our deep metric learning approach combined with SVM and PCA.

	Loss Function	**Precision**	**Sensibility**	**F1-Score**	**Specificity**
P. malarie	Circle	0.885 (0.088)	0.881 (0.075)	0.881 (0.072)	0.998 (0.002)
P. vivax	Circle	0.890 (0.044)	0.877 (0.038)	0.883 (0.030)	0.994 (0.003)
P. ovale	Multisimilarity	0.933 (0.084)	1.000	0.963 (0.046)	0.997 (0.003)
P. falciparum	Multisimilarity	0.837 (0.017)	0.918 (0.010)	0.982 (0.001)	0.976 (0.002)

Tabela 10: Model performance in *Plasmodium* parasite classification for dataset MP-IDB

The results are promising, particularly for P. ovale, which achieved a perfect sensitivity (1.000) and a high F1-score (0.963). P. malariae and P. vivax also showed consistent performance, with precision and sensitivity around 88%, and near-perfect specificity values of 0.998 and 0.994, respectively. For P. falciparum, although precision was slightly lower at 0.837, it still achieved strong results in terms of sensitivity (0.918) and F1-score (0.982). These results demonstrate the model's ability to accurately diagnose different species of Plasmodium, highlighting its effectiveness across varied species within the same parasite family.

We decided to explore a multiclass classification problem using the same pipeline to classify different types of parasites. The results from the ALL-DB dataset, shown in the table 11, indicate that our model performs well across various parasite species, including Babesia, Leishmania, Plasmodium, Toxoplasma, and Trypanosome.

	Loss Function	**Precision**	**Sensibility**	**F1-Score**	**Specificity**
Babesia	Circle	0.994 (0.010)	0.979 (0.030)	0.986 (0.010)	0.998 (0.001)
Leishmania	Circle	0.987 (0.007)	0.992 (0.010)	0.989 (0.006)	0.996 (0.006)
Plasmodium	Circle	0.983 (0.020)	0.967 (0.030)	0.975 (0.020)	1.000
Toxoplasma	Circle	1.000	1.000	1.000	0.995 (0.001)
Trypanosome	Circle	1.000	0.992 (0.006)	0.996 (0.003)	1.000

Tabela 11: Model performance in parasite classification for dataset ALL-DB

The model achieved high precision and sensitivity for all classes. Notably, Toxoplasma and Trypanosome both reached perfect scores of 1.000 for precision and sensitivity. Babesia and Leishmania also showed strong results, with precision values of 0.994 and 0.987, respectively, and sensitivity values close to 1.000. The Plasmodium species performed well too, with a sensitivity of 0.967 and precision of 0.983.

These results demonstrate that our model effectively handles multiclass classification and maintains high performance when distinguishing between different types of parasites.

5

Conclusion

The comparison of various deep metric learning methods and classifiers demonstrated significant potential for cytological data imaging applications. In addition to the metric learning approaches, we also compared the performance with convolutional models such as VGG19 and DeCAF, but the deep metric learning models, especially Circle Loss, consistently outperformed them. Circle Loss emerged as the best across all classification metrics, particularly excelling in sensitivity (98.3%) and specificity (99.3%).

Our experiments also extended to the classification of multiple parasite species, specifically, in the MP-IDB malaria dataset, our models reached a remarkable 0.98 recall. For Trypanosoma cruzi trypomastigotes in the TRYP-DB dataset, we achieved a sensitivity of 94%, showing the model's strong performance with this parasite. Additionally, the results in classifying different species of the Plasmodium parasite were impressive, with the model achieving a perfect 1.000 sensitivity in *P. ovale*, and high performance in other species as well. This highlights the model's ability to handle different species from the same parasite effectively.

Our multiclass experiments, which included various parasites like Babesia, Leishmania, Plasmodium, Toxoplasma, and Trypanosome, further demonstrated the robustness of our approach, with more than 96% sensitivity achieved for all categories. These results underscore the model's capability to generalize across a range of parasite species while maintaining high accuracy and sensitivity.

In summary, the research successfully compared multiple deep metric learning algorithms and demonstrated their strong performance in medical diagnosis. Circle Loss proved to be especially valuable, and the findings indicate promising directions for future research, including further refinement of models and exploration of additional parasites and imaging conditions.

References

[And] First and second principal component in a 3-covariates setting. https://bookdown.org/andreabellavia/mixtures/principal-component-analysis.html. Accessed: 2024-08-28.

[hvc] hvchealth echocardiogram description. https://www.hvchealth.com/echocardiogram8. Accessed: 2024-02-19.

[jan] janosh conv2d. https://tikz.net/janosh/conv2d.png. Accessed: 2024-02-19.

[upg] upgrad basic-cnn-architecture. https://www.upgrad.com/blog/basic-cnn-architecture/. Accessed: 2024-02-20.

[5] Abdi, H. and Williams, L. J. (2010). Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459.

[6] 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.

[7] 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.

[8] Bro, R. (2003). Multivariate calibration. *Analytica Chimica Acta*, 500(1–2):185–194.

[9] 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.

[10] Duan, Y., Lu, J., Feng, J., and Zhou, J. (2018). Deep localized metric learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2644–2656.

[11] 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.

[12] 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.

[13] Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., and Socher, R. (2021). Deep learning-enabled medical computer vision. *npj Digital Medicine*, 4(1).

[14] 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.

[15] 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.

[16] F.R.S., K. P. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

[17] 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).

[18] Gonzalez, R. C. and Woods, R. E. (2018). *Digital Image Processing*. Pearson.

[19] 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).

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

[21] 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.

[22] 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.

[23] 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.

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

[25] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

[26] Li, C., Ma, X., Jiang, B., Li, X., Zhang, X., Liu, X., Cao, Y., Kannan, A., and Zhu, Z. (2017). Deep speaker: an end-to-end neural speaker embedding system.

[27] Li, S. (2020). Microscopic images of parasites species.

[28] 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*.

[29] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis.

[30] Loddo, A., Di Ruberto, C., Kocher, M., and Prod'Hom, G. (2019). Mp-idb: The malaria parasite image database for image processing and analysis. In Lepore, N., Brieva, J., Romero, E., Racoceanu, D., and Joskowicz, L., editors, *Processing and Analysis of Biomedical Information*, pages 57–65, Cham. Springer International Publishing.

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

[32] Morais, M. C. C., da Silva, D. M., Oliveira, M. T., de Lana, M., and Nakaya, H. I. (2021). Image dataset from automatic analysis of trypanosoma cruzi: A machine learning approach for the detection of blood trypomastigotes in low resolution images.

[33] Morais, M. C. C., Silva, D., Milagre, M. M., de Oliveira, M. T., Pereira, T., Silva, J. S., Costa, L. d. F., Minoprio, P., Junior, R. M. C., Gazzinelli, R., de Lana, M., and Nakaya, H. I. (2022). Automatic detection of the parasite trypanosoma cruzi in blood smears using a machine learning approach applied to mobile phone images. *PeerJ*, 10(e13470):e13470.

[34] 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.

[35] O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks.

[36] Oyibo, P., Jujjavarapu, S., Meulah, B., Agbana, T., Braakman, I., van Diepen, A., Bengtson, M., van Lieshout, L., Oyibo, W., Vdovine, G., and Diehl, J.-C. (2022). Schistoscope: An automated microscope with artificial intelligence for detection of schistosoma haematobium eggs in resource-limited settings. *Micromachines (Basel)*, 13(5):643.

[37] Oyibo, P., Meulah, B., Agbana, T., Bengtson, M., van Lieshout, L., Oyibo, W., Vdovine, G., and Diehl, J.-C. (2023). Schistosoma haematobium egg image dataset.

[38] 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.

[39] 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).

[40] 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.

[41] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2014). Imagenet large scale visual recognition challenge.

[42] 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.

[43] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

[44] 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.

[45] 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.

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

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

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

[53] 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.