Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

GeMGF - Generic Multimodal Gradient-Based Meta Framework

GeMGF - Meta Framework Multimodal baseado em Gradiente

Liriam Michi Enamoto

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Computer Science) in The University of Brasilia

Brasília
2023

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

GeMGF - Generic Multimodal Gradient-Based Meta Framework

GeMGF - Meta Framework Multimodal baseado em Gradiente

Liriam Michi Enamoto

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Computer Science) in The University of Brasilia

Prof. Dr. Li Weigang (Advisor)
CIC/UnB

Prof. Dr. Geraldo P. Rocha Filho
UnB

Prof. Dr. Luis Paulo Faina Garcia
UnB

Prof. Dr. Jó Ueyama
USP

Prof. Dr. Paulo C. Costa
George Mason University

Prof. Dr. Ricardo Pezzuol Jacobi
Coordenador do Programa de Pós-graduação em Informática

Brasília, April , 2023

Dedication

I dedicate this thesis to my parents, who always encouraged me to pursue my goals, and to my husband, who has been a constant source of support during this challenging journey.

Acknowledgements

I would like to thank my advisor, Prof. Li Weigang, for his consistent support, guidance, generosity, and patience. His encouragement, motivation and valuable feedback were fundamental to the completion of this course.

I would like to extend my sincere thanks to Prof. Geraldo Rocha, who shared his expertise and experience, Prof. Alba Melo, Prof. Célia Ghedini, Prof. Genaína Rodrigues, Prof. Maristela Holanda, Prof. Maurício Ayala, Prof. Mylène Faria, Prof. Luís Garcia, and other members of PPGI.

I am also grateful to my colleagues at Translab and CIC/UnB for their friendship and support. I also could not have undertaken this journey without the understanding of my work colleagues.

Lastly, I am grateful to the University of Brasilia and Brazilian public education system that gives equal opportunities for all students willing to cooperate with scientific research.

Resumo

O surgimento do *Transformer*, um modelo pré-treinado utilizando dados em larga escala, e as suas recentes novas versões têm revolucionado as pesquisas de *Machine Learning* em linguagem de processamento natural e visão computacional. Os excelentes resultados obtidos pelos modelos baseados em *Transformer* dependem de dados rotulados de alta-qualidade e de um domínio específico em estudo. No entanto, devido à diversidade de situações em que esses modelos são utilizados, é desafiador criar modelos que aprendam a partir de um conjunto limitado de dados. O modelo pode apresentar falta de generalização, vieses de linguagem e falta de imparcialidade causados pelos modelos pré-trainados o que pode levar a resultados inesperados em aplicações do mundo real. Este problema não resolvido nos levou à pesquisar sobre *Multimodal Few-Shot Learning.*

Foi efetuada uma revisão sistemática abrangente na literatura em que 138 trabalhos publicados após 2019 sobre *Multimodal Few-Shot Learning* foram selecionados. Selecionamos 19 artigos finais divididos em dois grupos. O primeiro grupo é representado pelos modelos que utilizam um grande conjunto de dados para o treinamento (*Teacher Network*) e transfere o conhecimento adquirido para executar a tarefa principal (*Student Network*). Neste grupo, podemos citar como exemplo o *Transformer.* O segundo grupo utiliza diversos métodos: (i) aprendizado baseado em otimização; (ii) *Graph Neural Network (GNN)*; (iii) *Generative Adversarial Network (GAN)*; (iv) *Zero-Shot Learning (ZSL).* Uma análise detalhada sobre a metodologia, vantagens e desvantagens das abordagens de *Multimodal Few-Shot Learning* em cada um dos 19 artigos nos permitiu identificar os problemas ainda não endereçados.

As lacunas encontradas na revisão sistemática nos levou a desenvolver o **Ge**neric **M**ultimodal **G**radient-Based **M**eta **F**ramework **(GeMGF)**. Para compensar a falta de dados, utilizamos dados multimodais em que informações suplementares e complementares de uma modalidade podem auxiliar na representação dos dados. Os dados multimodais são extraídos utilizando modelos de *deep leaning* e então representados em um espaço vetorial unificado.

Abordamos o problema do aprendizado com poucos dados através de duas perspectivas: modelo e dados. Considerando a perspectiva do modelo, o algoritmo pode ter dificuldade de generalização no aprendizado supervisionado caso os dados nunca vistos utilizados no conjunto de teste não estiverem contidos no conjunto de treinamento. Este problema foi endereçado por meio do *meta-learning* em dois níveis de aprendizado: *base-learner* e o *meta-learner*.

Considerando a perspectiva dos dados, a falta de dados de treinamento foi compensado pelo aprendizado multimodal em que informações complementares de uma modalidade podem ajudar na representação dos dados. O principal objetivo do aprendizado multimodal é criar uma abstração da representação unificada das diferentes modalidades. A representação de dados multimodais apresenta alguns desafios dada a heterogeneidade da estrutura, tamanho e dimensão dos dados das diversas modalidades. Neste processo, a escolha do tipo de fusão multimodal é importante para permitir o alinhamento ou fusão entre os dados heterogêneos de cada modalidade.

Entrando em mais detalhes sobre a perspectiva do modelo, o GeMGF é composto pelo *base-learner* e o *meta-learner*. O *base-learner* é repensável pela extração e representação dos dados multimodais, composto por quatro sub-modelos: (i) *image embedding* (sub-modelo 1); (ii) *text embedding* (sub-modelo 2); (iii) *multimodal embedding* (sub-modelo 3); e (iv) *Multimodal Few-Shot Learning)* (sub-modelo 4). O *Residual Neural Network* (ResNet) foi utilizado para a extração de imagens por ser adaptável conforme a disponibilidade de recurso computacional. Utilizamos o ResNet30, contendo apenas 30 *identity blocks*. O *Bidirectional Long Short-Term Memory* (BiLSTM) foi utilizado para a extração de textos por permitir capturar o contexto do *time step* do passado e do futuro em textos longos. Após a extração dos dados, o modelo aprende o alinhamento entre imagem e texto integrando os dados em um mesmo espaço vetorial para reduzir o *gap* semântico entre as modalidades. Utilizamos a fusão a nível de decisão em que os dados de cada modalidade são extraídos separadamente e cada modalidade possui um classificador específico. Então o *Prototypical Network* e o *Relation Network* são utilizados para aprender a relação entre o protótipo de cada classe e os dados do *query set*.

O *meta-learner* é responsável por atualizar periodicamente os parâmetros do *base-learner* por meio do *Reptile* — um *meta-learner* baseado em otimização. O *Reptile* e o *Few-Shot Learning* (FSL) auxiliam a otimizar o aprendizado do framework, mesmo utilizando poucos dados para o treinamento. A configuração do GeMGF como um todo reduz a dependência de um dataset rotulado com grande volume de dados. Adicionalmente ao framework multimodal, criamos a versão unimodal para avaliar a sua flexibilidade e adaptabilidade em diferentes cenários.

O framework foi validado por meio de dez conjuntos de dados de diversas áreas: textos curtos do Twitter, textos longos da área jurídica, textos com caracteres alfabéticos (inglês e português) e não-alfabéticos (japonês), imagens da área médica e dados multimodais.

O framework unimodal para texto foi validado por meio de oito conjunto de dados, sendo cinco conjuntos de dados reais de diversas áreas (EN-T, Tweet250, JP-T, Livedoor e DEC6). Utilizamos também três conjuntos de dados *benchmark* para comparação (20NG, Oxford-102 e CUB-200-2011). Por meio dos experimentos, analisamos a dependência do framework da qualidade, quantidade, idioma do texto e distribuição dos dados entre as classes. O framework unimodal superou o modelo *baseline* em sete conjunto de dados (EN-T, Tweet250, JP-T, Livedoor, DEC6, CUB-200-2011 e Oxford-102), sendo que o GeMGF unimodal superou tanto o modelo *baseline* como o *Transformer BERT* com os conjunto de dados CUB-200-2011 e Tweet250. O framework unimodal para texto alcançou resultados excelentes com dados textuais em japonês, superando o modelo *Transformer BERT* em 58,30% com 90,90% menos parâmetros. Este excelente resultado sugere que a rica representação dos caracteres em japonês (*kanji*) auxiliou a criar um protótipo de classe de qualidade, porém é necessário uma investigação mais aprofundada para analisar o resultado.

O framework unimodal para imagem foi validado por meio de dois conjuntos de dados da área médica (COVID19 e Malaria) e dois conjunto de dados *benchmark* (Oxford-102 e CUB-200-2011). O GeMGF para imagem atingiu resultados similares ao modelo EfficientNet V2 somente com o conjunto de dados COVID19. O EfficientNet V2 se beneficiou do conhecimento adquirido no pré-treinamento utilizando ImageNet que possui 1,2 milhões de imagens de 1000 classes diferentes, inclusive flores e pássaros contidos nos conjuntos de dados Oxford-102 e CUB-200-2011.

O framework multimodal superou em 1,43% o modelo estado-da-arte de Munjal et al. 2023 com CUB-200-2011, e superou em 1,93% o modelo de Pahde et al. 2021 com Oxford-102. O resultado do framework multimodal foi 34,68% superior ao framework unimodal para imagem com CUB-200-2011, e 13,96% superior com Oxford-102. Os resultados sugerem que a combinação de dados textuais e imagens podem auxiliar no aprendizado e na melhoria da performance do framework como um todo.

Para analisar o impacto de quatro componentes do GeMGF, efetuamos as seguintes *ablation analyses*: (i) *Relation Network*; (ii) *image embedding* (sub-modelo 1); (iii) *text embedding* (sub-modelo 2); e (iv) tipo de fusão multimodal. O *Relation Network* foi o componente de maior impacto e foi validado por meio da substituição pela distância euclidiana. O framework obteve uma acuária 109,90% superior com o *Relation Network* quando comparado à distância euclidiana com CUB-200-2011 e 97,54% superior com Oxford-102. O resultado sugere que o *Relation Network* auxilia o modelo a aprender a relação entre o protótipo da classe e os dados do *query set* de forma mais eficiente.

O tipo de fusão multimodal foi o segundo componente de maior impacto. Ao substituir a fusão a nível de decisão pela fusão a nível de características, a acurácia do framework

diminuiu em 41,63% com CUB-200-2011 e 43,56% com Oxford-102. O resultado sugere que a escolha da fusão multimodal é um dos fatores chaves no aprendizado multimodal.

O terceiro componente de maior impacto no GeMGF foram os dados textuais, validados por meio do congelamento das camadas treináveis do *text embedding* (sub-modelo 2). Observou-se uma diminuição na acurácia de 45,10% com CUB-200-2011 e 36,92% com Oxford-201.

O componente de menor impacto no framework multimodal foram os dados de imagens, validados por meio do congelamento das camadas treináveis do *image embedding* (sub-modelo 1). Observou-se um decréscimo na acurácia de 5,15% com CUB-200-2011 e 7,46% com Oxford-201. Esse baixo impacto pode ser explicado pela arquitetura compacta do *image embedding* (sub-modelo 1) composto pelo ResNet30 contendo somente três milhões de parâmetros. A arquitetura deste sub-modelo poderia ser melhorado aumentando a profundidade do *ResNet* e utilizando conhecimento externo por meio de pré-trinamento, porém esta mudança acarretaria em um aumento no custo computacional.

O impacto ambiental causado pelo treinamento de modelos complexos tem chamado a atenção da comunidade acadêmica devido ao aumento das emissões de carbono proveniente de *data centers*. Muitos modelos de *machine learning* são treinados em serviços na nuvem, incluindo o nosso framework que foi treinado no Google Colab. Consideramos a preocupação de criar modelos pequenos e compactos bastante relevante, pois o treinamento desses modelos coletivamente podem contribuir para o aumento das emissões de carbono. Efetuamos a medição do consumo de recurso computacional do GeMGF por meio de dois fatores: o número de parâmetros treináveis e a quantidade de operações de ponto flutuante (FLOP). O GeMGF multimodal utiliza 14 milhões de parâmetros 99,8% a menos que o Multimodal Transformer.

As principais contribuições desta pesquisa são: (i) um novo framework FSL multimodal que reduz a degradação do modelo quando treinado com poucos dados; (ii) GeMGF é treinado sem utilizar o conhecimento externo evitando vieses de linguagem e a falta de imparcialidade; (iii) GeMGF possui extratores de dados multimodais independentes e flexíveis que podem contribuir para aumentar a sua aplicabilidade; e (iv) o GeMGF unimodal para texto pode ser adaptado para idiomas alfabéticos e não-alfabéticos com ótimos resultados.

Como trabalhos futuros, pretendemos melhorar o modelo nos seguintes aspectos: (i) fornecer transparência e confiabilidade nos resultados por meio de *Explainable Model*; e (ii) aprofundar a análise do modelo utilizando multi-idiomas, especialmente idiomas asiáticos.

Palavras-chave: Multimodal Learning, Few-Shot Learning, Meta-learning, Data Fusion

Abstract

The emergence of Transformer — a model pre-trained over a large-scale dataset — and the recent new versions have revolutionized research in Machine Learning, especially in Natural Language Processing (NLP) and Computer Vision. The excellent results of Tranformer-based models depend on labeled and high-quality domain specific data. However, due to the diversity of contexts in which these models are used, it is challenging to create models that learn from limited data. The model may suffer from a lack of generalization, language bias, and fairness issues caused by large pre-trained models, resulting in unexpected outcomes in real-world applications. This open problem leads to research in multimodal Few-Shot Learning (FSL).

In this thesis, we devised the Generic Multimodal Gradient-Based Meta Framework (GeMGF). To compensate for the scarcity of data, we use multimodal data in which supplementary and complementary information of one modality can help the data representation. The multimodal data are extracted using deep learning models and represented in a unified vector space. The framework uses the Prototypical Network and Relation Network in the FSL. The Reptile — an optimization-based meta-learner — helps avoid model degradation with unseen data. In addition to the multimodal framework, we created the unimodal version to evaluate the flexibility and adaptability of the framework in different scenarios.

The framework was evaluated using ten datasets from various domains and characteristics, including short texts from Twitter, legal domain long text, text with alphabetic (English and Portuguese) and non-alphabetic (Japanese) languages, medical domain images, and multimodal benchmark datasets. Our multimodal framework was evaluated using CUB-200-2011 and Oxford-102 datasets, outperforming the state-of-the-art model of Munjal et al. [1] by 1.43% with CUB-200-2011 and Pahde et al. [2] by 1.93% with Oxford-102. The result of the multimodal framework with CUB-200-2011 was 34.68% higher than the unimodal framework for image and 13.96% higher with Oxford-102. The results suggest that text and image data jointly helped the framework learn rich information and improve overall performance. The multimodal GeMGF is a simple and compact framework using only 14 million parameters, 99.8% less than the Multimodal Transformer. The unimodal framework for text achieved excellent results with the Japanese dataset, outperforming Transformer BERT by 58.30% with 90.90% fewer parameters. These results suggest that our framework achieved better performance with a significant computational cost reduction.

The main contributions of our research are: (i) a novel multimodal FSL framework, GeMGF is developed to reduce the model degradation trained over a few data; (ii) GeMGF is trained without external knowledge avoiding language bias and fairness issues; (iii) GeMGF has independent and flexible feature extractors that enhance its applicability; and (iv) the unimodal framework for text can be adapted to process alphabetic and non-alphabetic languages with high performance.

Keywords: Multimodal Learning, Few-Shot Learning, Meta-learning, Data Fusion

Contents

List of Figures

List of Tables

Acronyms

BERT Bidirectional Encoder Representations from Transformer.

BiLSTM Bidirectional Long Short-Term Memory.

CNN Convolutional Neural Network.

FLOP Floating Point Operation.

FSL Few-Shot Learning.

GAN Generative Adversarial Network.

GCN Graph Convolutional Network.

GeMGF Generic Multimodal Gradient-Based Meta Framework.

GNN Graph Neural Network.

LSTM Long Short-Term Memory.

MLP Multi Layer Perceptron.

NLP Natural Language Processing.

ResNet Residual Neural Network.

RGB Red, green and blue.

RNN Recurrent Neural Network.

SGD Stochastic Gradient Descent.

VAE Variational Auto-Encoder.

ZSL Zero-Shot Learning.

Chapter 1

Introduction

Machine learning algorithms have been used in many aspects of our lives, including improving user experience in online shopping [8], movie suggestion [9], and improving the judiciary system efficiency [4]. It also has been used in critical decision-making, in which the model's wrong prediction may cause direct financial loss, such as financial fraud detection [10, 11], investment risk identification [12], and others.

In a more sensitive area, the model's wrong prediction can lead to potential health threats, such as breast cancer prediction [13], drug tests [14], and autonomous vehicles [15]. Devising models that not only have high-performance metrics in a controlled test environment but models that provide robustness in the real-world scenario is essential to provide safety and reliability. However, due to the diversity of domain contexts in which these models are used, it is challenging to create models that learn from limited data, adapt, and generalize in the open-world scenario.

This open problem leads to research in multimodal Few-Shot Learning (FSL), where the model acquires new concepts from a few data composed of one or more modalities.

1.1 Motivation

When human beings need to learn a new task, they usually try to find a relation between the new task and some similar experience that they already had. Based on the amount of experience and knowledge accumulated, they can learn new tasks. Several traditional machine learning models [16, 17] that use previous knowledge have been developed. Deep learning models, such as Convolutional Neural Network (CNN) [18] and Recurrent Neural Network (RNN)[19] revolutionized machine learning. CNN inspired other deep networks, such as AlexNet [20], Inception [21], and EfficientNet [22], increasing the dependency on previous knowledge, meaning more data. With the emergence of Transformer, [23] pre-trained over large-scale public datasets and large knowledge databases, machine learning has made significant progress. This attention-based architecture boosted recent research on natural language processing (NLP) [24], computer vision [25], vision and language [26, 27], bio-medicine [28], and others.

When a limited amount of data is used to train the machine learning model, it can give excellent results in the training and testing phases. To achieve these results, most models depend on some premises: (i) large labeled dataset; (ii) closed-world assumption, where the model might be trained on a large dataset but still represent a limited sample of the real-world; (iii) high coverage and quality of training data [29].

If one of the premises is not met, the model may have unsatisfactory results with unseen data. The model may suffer from a lack of generalization, language bias and fairness issues caused by large pre-trained datasets, model outcome bias caused by imbalanced datasets, difficulty handling outliers, and unexpected results in real-world applications [29].

Humans can accomplish specific tasks with a few data by learning and adapting previous experiences. Similarly, machine learning models can learn from a few data. Using Few-Shot Learning (FSL), the model can be trained on a few samples of each class, learning a new task progressively. The advantage of FSL is that we can expose the model to a more realistic scenario where limited labeled data are available for training.

1.2 Problem Statement and Objectives of the Research

In this thesis, we consider the multimodal FSL to address the problem of creating models that learn from limited data, adapt, and generalize in the open-world scenario. In the literature, we can find several works that study this problem, such as: Transformer-based multimodal FSL that learns an optimized data alignment of different modalities [30, 31], optimization-based learning in which a meta-learner is used to help the model generalization [32], and an episodic projection scheme to construct a multimodal vector space using a few data [33]. However, there are still open issues to be addressed.

Considering a classification problem for image and text data, the model needs to be trained with a few samples of each class to address the cost of large labeled datasets. The model needs to learn from a multimodal dataset composed of image and text data and be evaluated with unseen instances of multimodal data. The raw image and text data are extracted and used in the learning mechanism. In this process, data from different modalities are heterogeneous in structure, size, and dimensions. To address the semantic gap among modalities, the model needs to learn the alignment between image and text data [34]. The knowledge learned from a few multimodal data needs to be preserved and adjusted during training, avoiding the previous knowledge being replaced by new knowledge. [35]. Some multimodal FSL models are complex to train, hard to adapt to other domains, and have high computational cost leading to high energy consumption [36].

In summary, we address the problems in multimodal FSL with the following general research objectives:

- Reduce the cost to annotate large datasets;
- Reduce the semantic gap between different modalities;
- Create a model that learns from a few data preserving previously learned knowledge;
- Create a compact model to reduce the growth of computational cost to train complex models.

1.3 Devised Model

In this thesis, we focus on machine learning models that generalize from a few available data for training. To this end, we designed the **Ge**neric **M**ultimodal **G**radient-Based **M**eta **F**ramework **(GeMGF)**. The framework is a set of models comprised of the base learner that aims to learn a specific task and a meta-learner that helps to improve the overall framework generalization. To compensate for the scarcity of data, we use a multimodal dataset in which supplementary and complementary information of one modality can help the data representation. The multimodal data are extracted using deep learning models and represented in a unified space. The framework uses FSL combined with meta-learning to avoid model degradation with unseen data. The learning process occurs continuously, acquiring new knowledge without forgetting previously learned experiences [32]

The framework is adaptable to different data extraction methods and domain contexts enhancing its applicability. Additionally, we created an unimodal version of the framework that can be used in various domains.

1.4 Contributions

Although the expressive improvement of machine learning algorithms in the last few years, most models still depend on clean and labeled data to achieve good results. Through this research, we expect the possible contributions including:

1. A novel multimodal framework with Few-Shot learning that can alleviate performance degradation trained over a limited and a few samples of data;
2. GeMGF is trained end-to-end from scratch, avoiding possible language bias and fairness issues of pre-trained models;
3. The framework has independent multimodal feature extractors adaptable to other architectures;
4. The framework has possibilities for applications in various domains;
5. The unimodal framework for text is multilingual and adaptable to alphabetic and non-alphabetic languages.

1.5 Organization of the Thesis

In this research, we analyze models that use multimodal data, describe methods for data extraction and integration, FSL algorithms, and problems that still need to be studied. A multimodal and unimodal framework are devised and evaluated with ten datasets. This thesis is organized as follows:

- Chapter 2 provides background about models that learn from a few data from two perspectives: model and data. From the model perspective, the main concepts of meta-learning and FSL are described. From the data perspective, the key points of multimodal data extraction, representation, and fusion are detailed;
- Chapter 3 describes the recent publications about multimodal FSL found in the literature. The publications were selected by protocols defined in the systematic literature review and divided into two categories: models with external knowledge represented by transfer learning and pre-trained models; and models without external knowledge based on methods to learn fast from a few data, such as optimization and data augmentation;
- Chapter 4 presents the devised model: the Generic Multimodal Gradient-Based Meta Framework (GeMGF). This chapter details how it addresses the research problems, describes the framework architecture and the datasets used in this work;
- Chapter 5 describes the implementation details of two variations of GeMGF: the multimodal and the unimodal framework. We also provide the hyperparameters setting, details about the tools and libraries used in the framework;

- Chapter 6 demonstrates the framework experiment results for multimodal GeMGF and unimodal GeMGF with three different data compositions: text data only, image data only, and multimodal data;
- Chapter 7 provides the ablation analysis of GeMGF by replacing or disabling the internal components, describes the computer resource consumption, and discusses some relevant aspects of our research;
- Chapter 8 summarizes the main contributions of our work, and indicates the future directions of this research.

Chapter 2

Background

This chapter describes the theoretical background related to our research. First, Section 2.1 describes an overview of Deep Learning models. Next, we approach the problem of model that generalizes from a few samples of data from two perspectives: data and model. Section 2.2 details the model perspective, describing the main concepts of meta-learning divided into two categories: metric-based meta-learning focused on Few-Shot Learning (FSL); and optimization-based meta-learning. Section 2.3 details the data perspective, describing the key points of multimodal data extraction, representation, and fusion.

2.1 Deep Learning

This section gives introductory concepts of four deep learning models: Subsection 2.1.1 describes the main characteristics of Convolutional Neural Network (CNN); Subsection 2.1.2 explores Recurrent Neural Network (RNN) in the scope of sequential models; Subsection 2.1.3 describes the advantage of Bidirectional Long Short-Term Memory (BiLSTM); and Subsection 2.1.4 explores the architecture of Residual Neural Network (ResNet).

2.1.1 Convolutional Neural Network (CNN)

Convolutional Neural Network [37] is a well-known deep learning architecture initially used for computer vision and is widely used in other domains, such as malware [38] and flood [39] detection. It is designed to learn the spatial features, such as edges, corners, textures, and shapes that best describe the image object.

Formally, Convolutional Neural Network consists of a sequence of one or multiple pairs of convolution and pooling layers. A convolution layer is composed of several computational units. Each computational unit takes as input a region vector that represents a small region of the input image, and the small regions collectively cover the entire data [40]. In Equation 2.1, a computational unit associated with the $l-th$ region of input x calculates the output o:

$$o = \sigma(W.r_l(x) + b) \tag{2.1}$$

where $r_l(x)$ is the input region vector that represents the $l-th$ region, W represents the weight matrix, b the bias, and σ represents a nonlinear activation function such as Rectified Linear Units (ReLU).

The matrix of weights W and the vector of biases b are learned through training, and they are shared by computational units in the same layer [40]. The output image of the convolution layer is passed to a pooling layer, which calculates the average or maximum value of each region [40]. The idea of the pooling layer is to capture the most relevant feature of each region.

CNN can be used for text data, as illustrated in Figure 2.1 [3]. In the input layer, each sentence of text data is transformed into a matrix of word embedding [41]. Word embedding is a distributed representation of words that reduce data sparsity problem [42] and can be trained as part of CNN training or adopt a pre-trained corpus such as Word2Vec [43]. The input layer is followed by two convolutional layers. Each convolution layer has a variable number of computational units, with each unit corresponding to a small region (one or more words) from the input text [44]. Similarly to CNN for images, CNN model for text can be composed of one or multiple pairs of convolution and pooling layers followed by a fully connected layer. The final output layer returns the prediction for the input text.

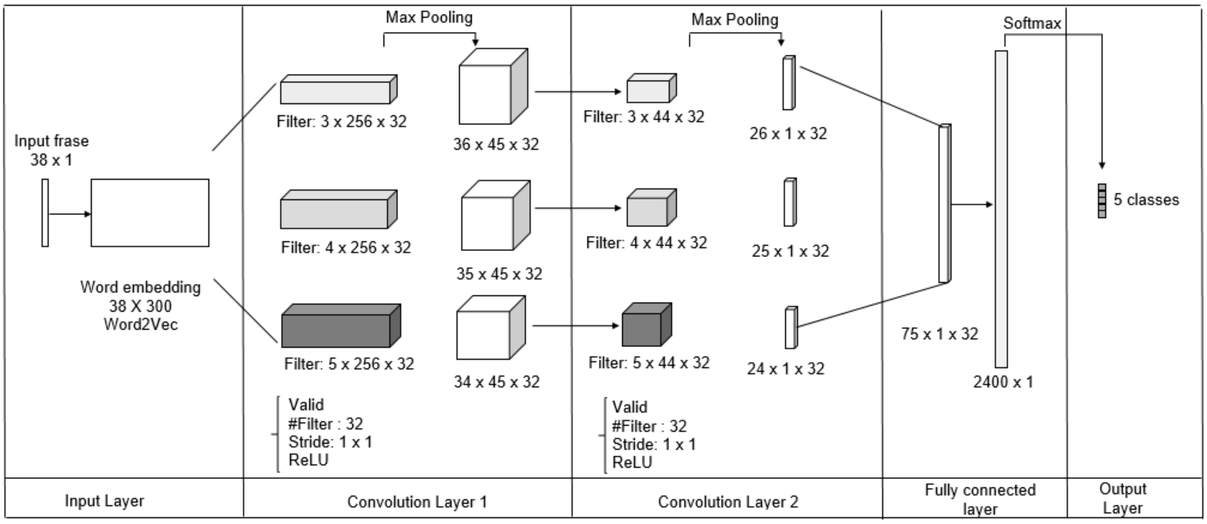


Figure 2.1: Example of CNN 2L architecture for short text classification [3].

2.1.2 Recurrent Neural Network (RNN)

Another widely used deep learning model is Recurrent Neural Network [45]. RNN is a sequential model architecture and can be applied to the Natural Language Processing (NLP) task where each word of a sentence is a time step. At each time step, it takes the current word and a hidden state from the previous time step as input and generates a new state. However, regular RNN architecture does not capture long-term dependencies between words due to vanishing gradient problems [46]. This problem is addressed by Long Short-Term Memory (LSTM) [47] by using hidden memory cell and gating units. It has three gates: input gate, forget gate, and output gate. Formally, the standard LSTM is expressed as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.2}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.3}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.4}$$

$$u = \tanh(W_u x_t + U_u h_{t-1} + b_u) \tag{2.5}$$

$$c_t = i_t \odot u + f_t \odot c_{t-1} \tag{2.6}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.7}$$

In the above equations, i_t, f_t and o_t are respectively input gate (2.2), forget gate (2.3) and output gate (2.4) at time step t. The input word at time step t is represented by x_t, W is the weight matrix for each gate, U the weight matrix for states and b represents the bias. σ is the *Sigmoid* function as a activation function of each gate. The symbol $\odot$ denotes element-wise multiplication. In order to generate the hidden state at current time step t, it generates a temporary result u (2.5) by *tanh* activation function over the input x_t and the preceding hidden state $h_{(t-1)}$. The hidden memory cell c_t (2.6) is updated by partially forgetting the existing memory and adding a new memory content. Finally, the hidden state h_t is calculated by equation (2.7) to be used in the following time step. In this way, LSTM detects an important feature from an input sequence at early stage and carries this information over long distance, capturing potential long-term dependencies.

2.1.3 Bidirectional Long Short-Term Memory (BiLSTM)

Regular LSTM capture the context of past input sequence but for long input data, it is interesting to capture the context of future time step as well. Bidirectional LSTM [48] address this issue by using two LSTM: one forward LSTM and one backward LSTM without the limitation of using input information just up to a present frame. The forward

LSTM processes the sequence from left to right. At each time step t, a hidden state is $\overrightarrow{h_t}$ computed based on the previous hidden state $h_{(t-1)}$ and the current step input x_t. The backward LSTM processes the sequence from right to left. At each time step t, a hidden state $\overleftarrow{h_t}$ is computed based on the future hidden state $h_{(t+1)}$ and the current step input x_t. Equation (2.8) represents new hidden state h_j computed by the concatenation of forward hidden state $\overrightarrow{h_t}$ and backward hidden state $\overleftarrow{h_t}$.

$$h_j = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{2.8}$$

By applying the BiLSTM, the whole long document T is processed forward and backward, capturing the context of a relevant word and keeping it for long-distance [49]. Figure 2.2 illustrates the BiLSTM model used in the legal domain for judicial long texts classification [4]. The input layer converts each word into a numeric vector. The BiLSTM layer process the forward and backward LSTM, followed by an attention layer. The output layer is responsible for the classification task.

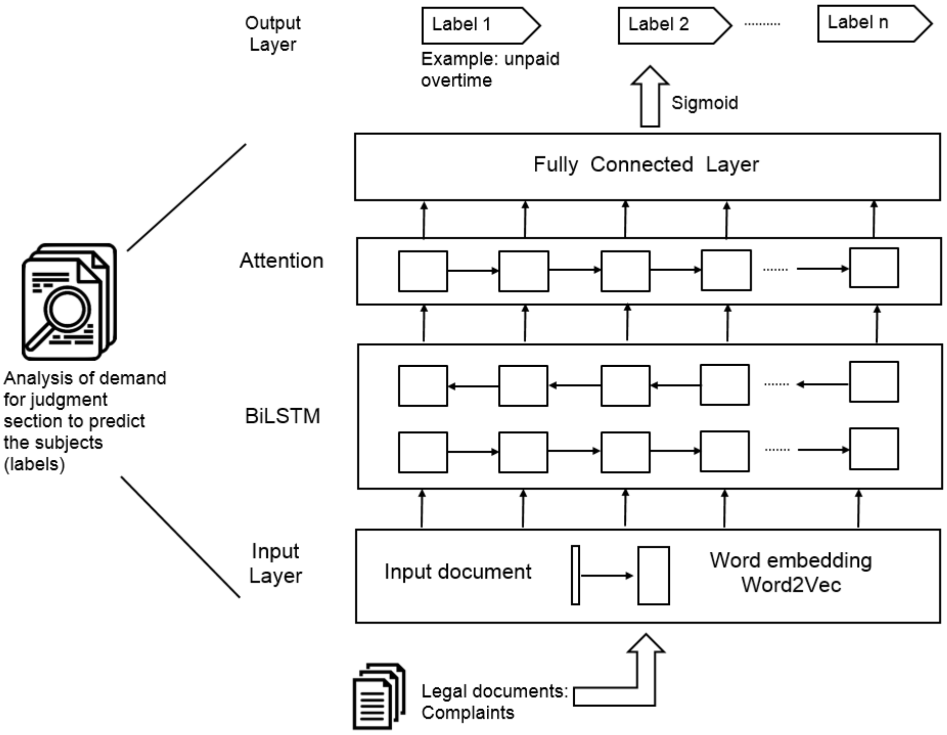


Figure 2.2: Example of BiLSTM architecture for long text classification [4].

2.1.4 Residual Neural Network (ResNet)

CNN models became widely used in computer vision boosting CNN-based deep networks, such as AlexNet [50], VGG-16 [51], Inception [21], ResNet [5], EfficientNet [22], and others.

Residual Neural Network (ResNet) [5] was proposed to address the degradation problem caused by the network depth increase. Instead of stacking several CNN layers, the model uses residual mapping, which is the main component of ResNet, illustrated in Figure 2.3. The input vector is represented by x, $\mathcal{F}(x)$ is the function that maps x to the output of the two weight layers, the curved arrow represented by x *identity* skips the two weight layers resulting in $\mathcal{F}(x) + x$.

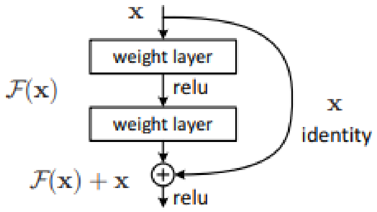

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.3: Residual mapping [5].

Formally, the residual block is defined in Equation 2.9, where x is the input vector, y is the output vector, and $\mathcal{F}(x, Wi)$ is the residual mapping.

$$y = \mathcal{F}(x, W_i) + x \tag{2.9}$$

The function $\mathcal{F}(x, Wi)$ can be implemented by multiple CNN layers. Concretely, ResNet comprises of several residual identity blocks and identity shortcut connections. Each identity block has two or more pairs of CNN layers, followed by a batch normalization layer [52]. The batch normalization layer normalizes the input of each layer for every mini-batch. The CNN layers calculate the residual information related to the input of each block, while the identity shortcut connections skip the CNN layers without adding extra parameters or calculations. The combination of several stacked identity blocks and identity shortcut connections helps to minimize the effects of vanishing gradient [46] and over-fitting, even for deep neural networks.

2.2 Meta-Learning

This section details the main ideas of meta-learning. In the regular supervised learning, the model is trained on a dataset $D = \{(x_i, y_i)\}_{i=1}^{m}$, where x_i is the input data and y_i is the output label. The model tries to learn the function f to map x_i to y_i, $f_\phi : x_i \rightarrow y_i$, where ϕ is the model parameter. The goal is to find the parameter ϕ when mapping x_i to y_i over the dataset D, such that the loss function $\mathcal{L}_D$ is minimized (Equation 2.10). This empirical risk minimization means that the model may not generalize well with real-world data if the unseen examples are not contained in dataset D [53].

$$\phi \leftarrow argmin_{\phi}\mathcal{L}_D(\phi) \tag{2.10}$$

Meta-learning, also known as “learn to learn” [54], studies how the algorithm can increase the ability to learn based on previous experiences. This learning mechanism is similar to how humans acquire knowledge and adapt to new tasks based on past experiences. We not only learn new concepts and skills but also learn how to generalize from a few examples.

Meta-learning address the limitation of regular supervised learning by adopting two learning levels: a meta-level and a base-level. The base-level can be a supervised learning model that aims to optimize a specific task. At this level, the model bias is calculated considering the relation of individual data points of the task. The meta-level aims to learn a set of tasks capturing the task structure variations by learning the entire function space. At the meta-level, the bias calculation considers the relatedness of the different tasks. Periodically, the meta-level model updates the parameters of the base-level model to improve generalization [55].

In the Equation 2.11, ϕ_{meta} is the meta-level parameter, $\mathbb{E}$ denotes the empirical expectation, $\mathcal{T}_j$ is a specific task, $p(\mathcal{T})$ is the probability of task distribution and $\mathcal{L}_{\mathcal{T}_j}$ is the loss of $\mathcal{T}_j$. The base-level model aims to minimize the loss $\mathcal{L}$ for a task $\mathcal{T}_j$, and the goal of the meta-level is minimize the loss over all tasks [53].

$$\phi_{meta} \leftarrow argmin_{\phi_{meta}}\mathbb{E}_{\mathcal{T}_j \sim p(\mathcal{T})}[\mathcal{L}_{(\mathcal{T}_j)}] \tag{2.11}$$

Meta-learning aims to improve the model prediction for the task $\mathcal{T}_j$ based on the knowledge extracted from different distributions of task $\mathcal{T}$ by using this two learning levels mechanism [56].

In the literature, we can find three categories of meta-learning: (i) metric-based [6, 7]; (ii) optimization-based [57, 58]; and (iii) model-based methods [59, 60, 61]. The first two categories are used in this research and are detailed in the following subsections.

2.2.1 Metric-Based Meta-Learning

This model type assumes that the features learned from data can be generalized, for instance, to calculate the distance between two images of an unknown class. The metric-based meta-learning is represented by Few-Shot Learning models.

Few-Shot Learning (FSL)

The FSL was first proposed by Weigang [62] and aims to learn a task from a few examples per class [63, 64]. In this method, the data comprise of support set and query set. The support set is used for model training and the query set for testing. The support set contains K samples of each C categories or classes. We use *C-way K-shot* notation to represent the classification task of C classes with K samples each. The advantage of using FSL is that we can expose the model to a more realistic scenario where only a few samples of labeled data are available for training.

To overcome the lack of data, the FSL model needs some special techniques to learn and generalize from a few data. For example, Siamese Network [65] is a non-linear metric-learning model that learns the similarity between a pair of objects; and Matching Network [66] uses attention mechanism and external memory to help the learning process.

FSL has been used in combination with Deep Learning [67, 7, 66, 6], where the support set is used to learn the embedding space, also known as metric learning. Then the learned metrics are used to predict the query set.

Next, we describe two FSL architectures used in this research: Prototypical Network and Relation Network.

Prototypical Network

Prototypical Network [6] uses a prototype representation for each class. It assumes that all data belonging to the same class cluster around a single point in the feature space: the class prototype. In this approach, the model learns an embedding space based on a neural network, and the class prototype is the mean vector of the support set projection in the embedding space.

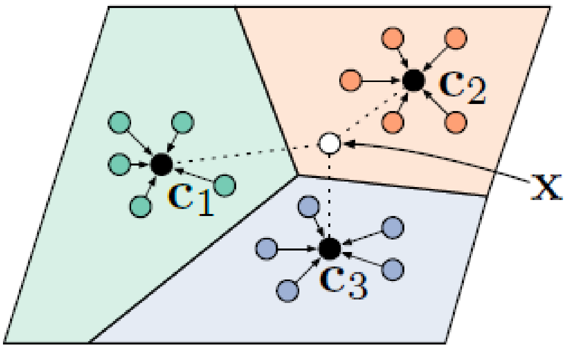


Figure 2.4: Prototypical Network [6].

Given a support set S with n labeled samples, $S = \{(x_1, y_1), ...,(x_n, y_n)\}$, each x_i is a vector, y_i is the corresponding label, and S_c is the labeled samples with c classes. The class protype $ClassP_c$ is calculated by Equation 2.12, where f_φ denotes the embedding function, such as a neural network, and x_i represents the elements of the support set S.

$$ClassP_c = \frac{1}{|S_c|} \sum_{(x_i,y_i)\in S_c} f_\varphi(x_i) \tag{2.12}$$

Similarly, the query set data are transformed via f_φ and the distances between query set data point and each class prototype are calculated by Bergman divergences [68], such as Euclidean distance. The probability distribution of one query point x belonging to each class is calculated by the softmax function over the distance d (Equation 2.13):

$$p_\phi(y = c|x) = \frac{exp(-d(f_\varphi(x), ClassP_c)}{\sum_{c'} exp(-d(f_\varphi(x), ClassP_{c'}))} \tag{2.13}$$

The model is trained via Stochastic Gradient Descent (SGD) to minimize the loss that represents a negative log-probability (Equation 2.14):

$$J(\phi) = -\log p_\phi(y = c|x) \tag{2.14}$$

Relation Network

Relation Network [7] is composed of two functions: an embedding function f_φ and a relation function g_ϕ. The embedding function is used to extract the features from the support set producing the future map $f_\varphi(x_i)$, and the features from the query set producing $f_\varphi(x_j)$. Then the feature maps $f_\varphi(x_i)$ and $f_\varphi(x_j)$ are concatenated by the function Z (Equation 2.15):

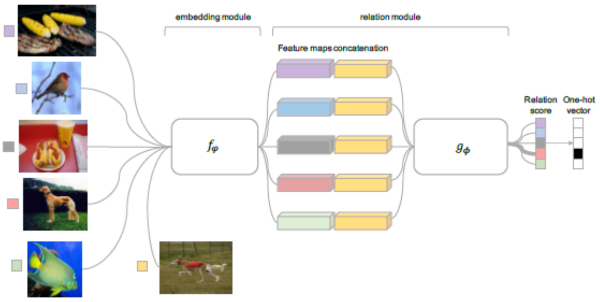


Figure 2.5: Relation Network [7].

$$Z = (f_\varphi(x_\mathrm{i}), f_\varphi(x_\mathrm{j})) \tag{2.15}$$

The purpose of the concatenation is to learn how the support set is related to the query set. This relation measure between x_i e x_j is calculated by the relation function g_ϕ, represented in Equation 2.16.

$$r_{ij} = g_{\phi}(Z(f_{\varphi}(x_i), f_{\varphi}(x_j))) \tag{2.16}$$

The relation function g_{ϕ} returns the score r_{ij} with values in the range of 0 to 1 representing the similarity between x_i e x_j. Although it is a classification problem, Sung et. al[7] used Mean Squared Error (MSE) as a loss function, since the label space $\in$ {0,1}. The model is trained to minimize the loss represented in Equation 2.17, where φ and ϕ denote parameters of the functions f and g, respectively:

$$\varphi, \phi \leftarrow argmin_{\varphi,\phi} \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{i,j} - 1(y_i == y_j))^2 \tag{2.17}$$

2.2.2 Optimization-Based Meta-Learning

The second type is optimization-based meta-learning, where the base-level task is solved as an optimization problem calculating the gradient to minimize the inner-loop loss. Then the meta-level model uses the error signals of the base-level to improve the overall model results and generalization ability [60]. Next, Reptile, a simple but efficient optimization-based algorithm used in this research is described.

Reptile

Reptile [57] is a gradient-based meta-learning algorithm. It learns an initialization value for the model parameters and generalizes from a small number of sample tasks at test time. Considering the model's initial parameter ϕ, for each iteration, the task τ is performed generating the loss L_{τ} which is minimized by gradient descent. The parameter ϕ is updated by Equation 2.18, where U_{τ}^{q} is the gradient descent operation that updates ϕ q times using data sampled from the task τ:

$$\tilde{\phi} = U_{\tau}^{q}(\phi) \tag{2.18}$$

Let τ_1 be the first task and the model performs Stochastic Gradient Descent (SGD) for q iterations to get the optimal parameter $\tilde{\phi}_1$. τ_2 is the second task, $\tilde{\phi}_2$ the optimal parameter value after SGD. Then the model tries to find the optimal value of the parameter ϕ, moving ϕ closer to $\tilde{\phi}_1$ and $\tilde{\phi}_2$. This is represented by Equation 2.19 calculating the gradients ∇_{ϕ} over the distance between ϕ and $\tilde{\phi}$, where $\tilde{\phi} = \{\tilde{\phi}_1, \tilde{\phi}_2\}$. α is the learning rate, and d is a distance function, such as Euclidean distance.

$$\phi = \phi - \alpha \nabla_{\phi} \frac{1}{2} d(\phi, \tilde{\phi})^2 \tag{2.19}$$

Then the randomly initialized parameter ϕ is updated in a direction closer to the optimal parameter $\tilde{\phi}$ by Equation 2.20. ϵ is a step size initialized with a fixed value and adjusted during training:

$$\phi = \phi + \epsilon(\tilde{\phi} - \phi) \tag{2.20}$$

This searching operation for the optimal parameter value avoids over-fitting and helps model generalization, even using a few samples of data.

2.3 Multimodal Learning

This section describes the main aspects of multimodal learning. The term modality represents a specific form in which data are available. In the past decade, several machine learning architectures have been used to successfully represent one modality, such as text, image, sound, video, and others. However, we capture multiple modalities signals from the surrounding world: we visualize images, read texts, hear sounds, feel textures and temperatures, and so on [69]. In some activities, we focus our concentration in one modality. For example, when we listen to a music, our brain process the audio information. However, in other more complex activities, different forms of representation help us better understand the context. For example, when two persons are having a conversation, the facial and body expression complement and enrich the audio information exchanged during the conversation. In this sense, the data comprised of more than one modality is known as multimodal data.

Multimodal learning aims to use supplementary and complementary information of the different modalities to execute one or more related tasks. There are some key points to consider before use this learning technique in machine learning: (i) how to extract and represent the multimodal data (Subsection 2.3.1); a (ii) how to align data from different modalities (Subsection 2.3.2). We discuss these topics in the following subsections.

2.3.1 Multimodal data extraction and representation

The multimodal data representation is challenging because of the heterogeneity of data structures, sizes, and dimensions. Some modalities, such as text, have a symbolic representation while video and audio have a signal representation. Usually, some initial treatment is performed before the multimodal data unification, and one alternative is to use deep learning models [69]. For instance: Convolutional Neural Network (CNN) for image feature extraction. CNN is a hierarchical architecture with a sequence of one or multiple pairs of convolution and pooling layers. Several filters convolve over the input matrix to extract the most significant features and predict the output efficiently [18]. Long Short-Term Memory (LSTM) [47] and the variation Bidirectional Long Short-Term Memory (BiLSTM) [48] are successfully used in text data extraction. LSTM captures the context of past input sequence while BiLSTM processes the input data forward and backward, capturing the context of a relevant word from the past and the future. The features extracted from multiple modalities represented by numeric vectors can be combined with some function to produce a new representation: the multimodal embedding data [69].

Formally, let's consider the multimodal dataset D comprised of two modalities, such as image and text. D contains a set of text annotations X, a set of images V and a set of labels Y. The text annotations can be denoted by $X = \{x_1,...,x_n\}$ where x_i is a single text annotation in X represented by a numeric vector. In this vector, each element corresponds to a word. Similarly, the image set can be represented as $V = \{v_1,...,v_n\}$ where v_i is a single image in V represented by a numeric vector, where each element corresponds to an image data point. The discrete label space can be represented as $Y = \{y_1,...,y_n\}$. In this context, the multimodal dataset D is denoted by $D = \{s_1,...,s_n\}$. Each s_i contains the tuple $\{(v_i,\ x_i,\ y_i)\}_{i=1}^{n}$ in which v_i represents a single image, x_i the annotation text that describes v_i, and y_i corresponds to the class label of v_i.

2.3.2 Multimodal data fusion

The main goal of multimodal learning is to create an abstraction of a unified representation of different modalities for each tuple s_i in $D = \{s_1,...,s_n\}$ and perform one or more tasks efficiently [70]. In this process, the heterogeneous multimodal data need to be integrated to find the relationship between two or more modalities, known as multimodal fusion [69].

In the literature [69, 71, 72], we can find three types of multimodal fusion methods: (i) late fusion or decision-level fusion; (ii) early fusion or feature-level fusion; and (iii) hybrid fusion or intermediate-level fusion.

Decision-level fusion

In this type of fusion, initially, the data from each modality are individually processed based on the decision task, such as classification. Then the modal data are integrated into the same feature space, as illustrated in Figure 2.6. The late fusion technique is more flexible because each modality has its classifier and predictor [69]. Therefore, the modal data can have different sampling rates or dimensions [71]. For instance, the image data can be extracted using CNN, and the audio data can be extracted by a feed-forward network. Each modality is processed by a specific decision making task. Then both feature vectors are concatenated in a unified representation [73].

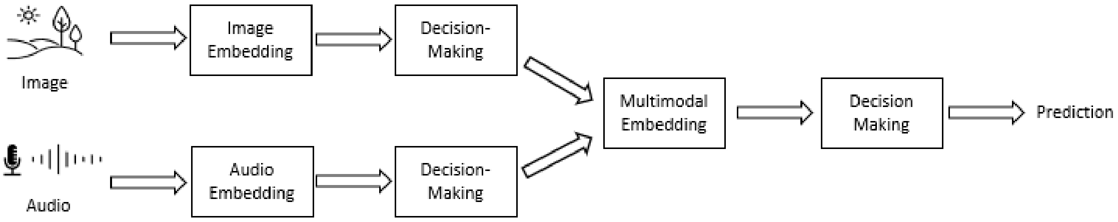


Figure 2.6: Decision-level fusion.

Feature-level fusion

The feature-level fusion exploits the low-level features of each modality just after the extraction, creating a strong interaction between modalities, as illustrated in Figure 2.7. A typical example is Transformer [23] which can be used to extract, represent, and learn optimized interaction between modalities, such as image and text [70]. The main drawback of this method is that the early fusion is performed on raw data, where features with different sampling rates and dimensions are extracted. For this reason, a large amount of data can be removed to perform the fusion of these modalities [71].

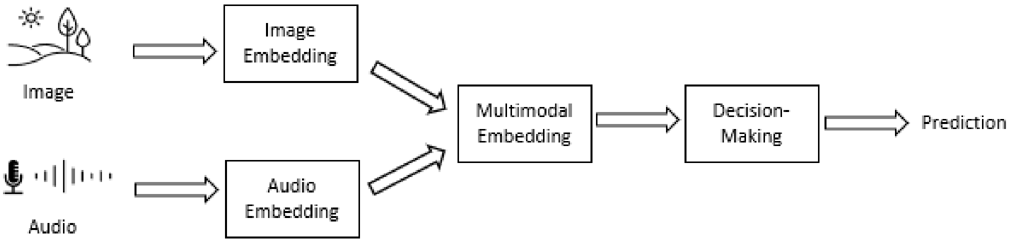


Figure 2.7: Feature-level fusion.

Hybrid fusion

Hybrid fusion or intermediate-level fusion learns a joint representation of different modalities by combining the decision-level and feature-level fusion. The fusion takes place at the commonly shared representation layer. The feature-level fusion contributes with low-level features representation and the decision-level fusion with high-level features that jointly help the model to learn a gradual fusion [71]. For example, data from various modalities such as facial expressions, galvanic skin response, and electroencephalogram can be extracted with different fusion levels to perform the emotion recognition task [72].

2.4 Chapter Summary

This chapter describes the theoretical background related to this research. First, an introduction to deep learning in which the widely used CNN, RNN, BiLSTM, and ResNet are described. Then, the background related to models that generalize from a few samples of data are presented. We can approach this problem from two perspectives: data and model. Considering the model perspective, the model is trained on a limited amount of data in the regular supervised learning. The goal is to minimize the loss which can be accomplished after a certain training epochs. However, the model may not generalize well if the unseen examples are not contained in the training dataset. The ability to learn and adapt quickly to new concepts is limited [53]. This problem can be addressed with meta-learning that uses two learning levels: a meta-level and a base-level. At the base-level, FSL can be used to train a model with a few samples of data. At the meta-level, optimization-based meta-learning can be used to periodically update the parameters of the base-level model and improve the overall model generalization.

Considering the data perspective, multimodal learning can be used to improve the model performance, where the model uses supplementary and complementary information from the different modalities to execute one or more related tasks. Deep learning models can be used for data extraction. However, multimodal data usually have different dimensions, sizes, and structures and need to be aligned before being compared to find similarities. The decision-level, feature-level, and hybrid fusion methods can be used for data alignment.

Chapter 3

Related Works

This chapter describes the recent works found in the literature about Multimodal Few-Shot Learning. To this end, we conducted a systematic literature review described in the following sections: Section 3.1 details the review protocol used in the systematic review; Section 3.2 describes the selected works using the defined protocols; Section 3.3 identifies and details the limitations and gaps in the selected works; and Section 3.4 presents the summary and considerations of this chapter.

3.1 Systematic Literature Review

The main objective of this systematic review is to identify state-of-the-art publications in multimodal models that learn from a few data. We identified six key points to guide our review: (i) the modality type; (ii) the embedding function used in the data extraction; (iii) the energy function used to calculate the distance between data points; (iv) the loss function; (v) the FSL method; and (vi) the multimodal data fusion type.

The following subsections detail the protocol used in the systematic review: Subsection 3.1.1 describes the keywords, the scientific digital libraries, and the eligibility criteria adopted; and Subsection 3.1.2 summarizes the result of the systematic review.

3.1.1 Review Protocol

To conduct a broad search in the literature, we selected some keywords and recent machine learning methods that have been explored with multimodal learning. The search terms and the scientific digital libraries defined in the review protocol are detailed in Table 3.1.

Search Terms	Scientific Digital Libraries	URL
Multi modal learning	Google Scholar	http:/scholar.google.com.br
Cross modal learning	Dblp	http:/dblp.org
Meta learning	Science Direct	http:/www.sciencedirect.com
Multi task learning	ACM Digital Library	http:/dl.acm.org
Deep learning	Springer Link	http:/link.springer.com
Few shot learning	IEEE Xplore Digital Library	http:/ieeexplore.ieee.org
Zero shot learning		

Table 3.1: Review protocol: search terms and scientific digital libraries.

The inclusion and exclusion eligibility criteria for recent works selection are detailed in Table 3.2.

Inclusion Criteria	Criteria description
IC1	Works that detail the model architecture and hyperparameters.
IC2	Research that use benchmark datasets.
IC3	Models that use FSL techniques.
IC4	Peer-reviewed papers.
Exclusion Criteria	**Criteria description**
EC1	Works with similar architecture and contributions.
EC2	Works written in a language other than English.
EC3	Works published before 2019.

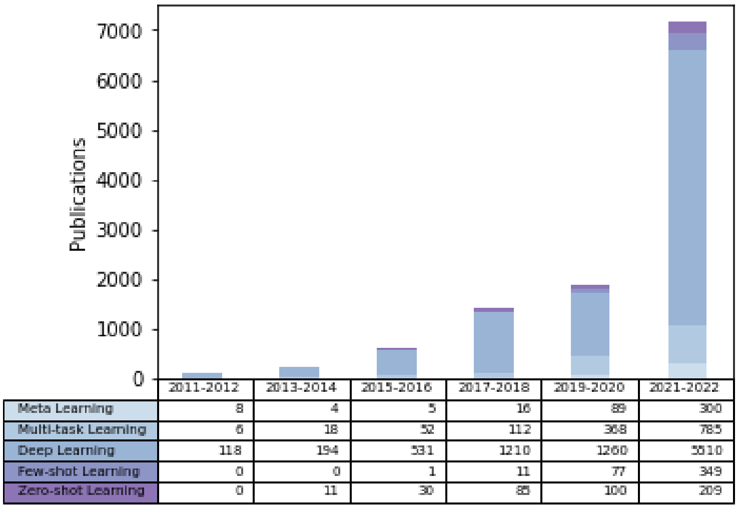
Table 3.2: Review protocol: eligibility criteria.

3.1.2 Review Execution

The attention of the academic community to multimodal learning has grown fast in the last years. We searched for the combination of the keywords described in Table 3.1 on Google Scholar, resulting in approximately 13,000 publications without date restrictions. The chart in Figure 3.1 shows the number of publications after 2011 in combination with the search terms. The bars represent publications about multimodal learning combined with meta-learning, multi-task learning, deep learning, few-shot learning, or zero-shot learning by biennium. We can observe exponential growth (270%) in the last biennium (2021-2022) compared with the previous biennium (2019-2020). The interest in the multimodal models with deep learning methods represents the majority (77%) of the publications. The publications about multimodal learning with FSL represent only 4% in 2021-2022 suggesting opportunities for further research in this field.

After having this overview of publications about multimodal learning since 2011, we conducted a systematic literature review using the open-source research tool Zotero [1]. First, the eligibility criteria IC4, EC2, and EC3 described in Table 3.2 were applied,

[1] https:/www.zotero.org



	2011-2012	2013-2014	2015-2016	2017-2018	2019-2020	2021-2022
Meta Learning	8	4	5	16	89	300
Multi-task Learning	6	18	52	112	368	785
Deep Learning	118	194	531	1210	1260	5510
Few-shot Learning	0	0	1	11	77	349
Zero-shot Learning	0	11	30	85	100	209

Figure 3.1: Number of publications of multimodal learning combined with one of the techniques : meta-learning, multi-task learning, deep learning, few-shot learning or zero-shot learning.

Inclusion Criteria	**Selected Publications**	
Peer-reviewed papers with hyperparameters (IC1, IC4)	Eloff et al.,2019[73]	Passalis et al.,2021[74]
	Islam el al.,2019[75]	Wang et al.,2020[70]
	Yu et al.,2020[76]	Li et al.,2021[77]
	Ji et al., 2022[78]	Li et al., 2021[79]
	Fang et al., 2022[80]	Fan et al., 2022[33]
	Zhu et al., 2022[31]	Munjal et al., 2023 [1]
Peer-reviewed papers with benchmark datasets (IC2, IC4)	Zhao et al.,2021[81]	Pahde et al.,2021[2]
	Eloff et al.,2019[73]	Song et al.,2020[32]
	Passalis et al.,2021[74]	Islam el al.,2019[75]
	Tonge et al.,2019[82]	Wang et al.2020[70]
	Yu et al.,2020[76]	Li et al.,2021[77]
	Bendre et al.,2021[83]	Tsimpoukelli et al.,2021[30]
	Ji et al., 2022 [78]	Li et al., 2021[79]
	Fang et al., 2022[80]	Fan et al., 2022[33]
	Zu et al., 2022[31]	Munjal et al., 2023 [1]
Peer-reviewed papers with FSL (IC3, IC4)	Zhao et al.,2021[81]	Pahde et al.,2021[2]
	Eloff et al.,2019[73]	Passalis et al.,2021[74]
	Ding et al.,2021[84]	Yu et al.,2020[76]
	Pan et al.,2020[85]	Tsimpoukelli et al.,2021[30]
	Ji et al., 2022 [78]	Li et al., 2021[79]
	Fan et al., 2022[33]	Zu et al., 2022[31]
	Munjal et al., 2023 [1]	

Table 3.3: Selected publications from systematic literature review.

resulting in 138 publications. Next, we manually applied the eligibility criteria IC1, IC2, IC3, and EC1, resulting in 19 publications. The details of each selected work are described in the following section.

3.2 Selected Works

This section details the works selected after the review execution described in Subsection 3.1.2. The multimodal Learning using FSL can be categorized into two groups: (i) multimodal FSL with external knowledge and (ii) multimodal FSL without external knowledge.

Table 3.4 summarizes the selected works, where the first column refers to the publication, and the following columns represent the six key points used in the literature review. The column 'Modality' refers to the modality type (A: audio; I: image; T: text; and V: video), the column 'Embedding Function' refers to the method used in the data extra extraction, the column 'Energy Function' represents the function used to calculate the distance between data points. The column 'Loss Function' represents the objective function used in the model optimization, and the column 'FSL' refers to the FSL method (P: Prototypical Network; S: Siamese Network; M: Matching Network; and Z: Zero-Shot Learning). The column 'Fusion' refers to the multimodal data fusion type, divided into D: decision-level fusion; F: future-level fusion; H: hybrid fusion, and N/A for unimodal data when there is no modality fusion.

The first five publications [73, 81, 70, 30, 31] are related to multimodal FSL with external knowledge and the remaining publications [32, 84, 86, 77, 2, 79, 83, 75, 80, 76, 85, 74, 33, 1] does not use external knowledge.

The selected works are detailed in the following subsections according to this categorization.

3.2.1 Multimodal FSL with external knowledge

When human beings need to learn a new task, they usually try to find a relation between the new challenge and some similar experience that they already had in the past. Based on the amount of experience or knowledge humans have, they can learn new tasks. Similarly, a pre-trained model with a large and multimodal dataset can be beneficial if the labeled data are scarce in the downstream task. In this method, the Teacher Network is trained over massive amounts of labeled data. The knowledge acquired from the Teacher Network is then transferred to the Student Network, which makes predictions based on a few samples to mimic FSL [81].

In multimodal learning, the data fusion type plays an important role. We selected two works that use external knowledge with decision-level fusion. In this fusion type, the different modalities are extracted separately with independent classifiers and then integrated into the same feature space. Eloff et al. [73] used a Convolutional Neural Network (CNN) for image and a feed-forward network for audio extraction in the Teacher Network.

Publication	**Modality** (*)	**Embedding Function**	**Energy Function**	**Loss Function**	**FSL** (**)	**Fusion** (***)
Multimodal FSL with external knowledge						
Eloff et al. 2019 [73]	I,A	CNN	Cosine similarity	Hinge loss	S	D
Zhao et al. 2021 [81]	I,T	ResNet-18, BiLSTM	Cosine similarity	Spatial Relation loss, Cross-entropy loss	P M	D
Wang et al. 2020 [70]	I,T	Transformer ResNet BiLSTM	-	Cross-entropy loss	-	H
Tsimpoukelli et al. 2021 [30]	I,T	Transformer ResNet	-	-	-	F
Zhu et al. 2022 [31]	I,T,V	Transformer	Cosine similarity	-	-	F
Multimodal FSL without external knowledge						
Song et al. 2020 [32]	I,T	CNN	-	Cross-entropy loss	-	D
Ding et al. 2021 [84]	T	GCN	Euclidean distance	Avr.negative loss	-	N/A
Ji et al. 2022 [78]	I,T	GNN ResNet-12	Euclidean distance	Cross-entropy loss	P	D
Li et al. 2021 [79]	I,T	GCN CNN	Cosine Similarity	Cross-entropy loss	-	D
Pahde et al. 2021 [2]	I,T	ResNet-18, GAN	Nearest neighbour	GAN loss	P	D
Li et al. 2021 [77]	I,T	ResNet-101 GAN	-	Task loss	Z	F
Bendre et al. 2021 [83]	I,T	ResNet-101 VAE, MLP	-	Multimodal loss	Z	F
Islam el al. 2019 [75]	A,T	CNN	Euclidean distance	Neighbour aware loss	Z S	F
Fang et al. 2022 [80]	I,T	ResNet VAE	-	MA loss	Z	F
Yu et al. 2020 [76]	I,T	ResNet-101 CNN-RNN	Euclidean distance	Cross-entropy loss	P Z	D
Pan et al. 2020 [85]	I,T	VGG-16	Nearest neighbour	Cosine dist. loss	P Z	D
Passalis et al. 2021 [74]	I	ResNet-101 CNN	Minimum distance	Centroid-based loss	P	N/A
Fan et al. 2022 [33]	I,T	ResNet-18 MLP	Cosine, Euclidean	-	-	D
Munjal et al. 2023 [1]	I	ResNet	-	Cross-entropy loss	S	N/A

(*) Modality - A: audio. I: image; T: text; V: video. (**) FSL - P: Prototypical Network; S: Siamese Network; M: Matching Network; Z: Zero-Shot Learning. (***) Fusion type - D: decision-level fusion; F: future-level fusion; H: hybrid fusion; N/A - not applicable for unimodal data. Loss - MA: mutual alignment.

Table 3.4: Selected works by systematic literature review.

Then, the knowledge was transferred to the Student Network composed of the Siamese Network. This model used cosine distance to calculate the similarity between images and dynamic time warping for audio. However, the successive unimodal comparison using different metric functions can lead to model degradation. One of the possible reasons for the degradation is that it is challenging to represent and compare data from different modalities, which is called the heterogeneity gap [69]. The model used intra-class and inter-class distances between data to apply different loss functions and reduce the overall error.

Zhao et al.[81] also trained the Teacher Network with decision-level fusion. The Prototypical Network and the Matching Network were used in the unimodal Student Network. The quality of embedding space is measured by inter-class and intra-class relationship. Different loss functions were applied depending on the measured embedding quality.

Next, we describe three works that use feature-level fusion and hybrid fusion. The feature-level fusion learns the interaction between modalities during the low-level feature extraction process, creating a strong data alignment. Hybrid fusion is the combination of decision-level and feature-level fusion. These two multimodal fusion types can be found in models that use Transformer. Wang et al. [70] proposed the multimodal Transformer pre-trained on ImageNet and based on hybrid fusion. The main network performs a decision-level fusion extracting image and text data separately. The meta-network performs a feature-level fusion with Transformer creating a unified feature space from image and text data. The limitation of this work is that the model may suffer from over-fitting caused by the multi-head attention mechanism.

Tsimpoukelli et al. [30] proposed a Transformer based multimodal few-shot learner. The feature-level data fusion is used to encode images into the word embedding space. The downstream task does not use any known FSL technique. Instead, the model relies on the knowledge accumulated by Transformer's 7 billion parameters to perform several tasks from a few data, such as image captioning.

Zhu et al. [31] used modality-agnostic Bidirectional Encoder Representations from Transformer (BERT) to process a variety of modalities and tasks into a unified representation space. Similarly to [30], the FSL in the fine-tuning phase was possible by the large-scale datasets used in the pre-training phase.

Transformer-based multimodal FSL has made great advances by handling a variety of modalities and tasks, learning an optimized data alignment, and helping to reduce the heterogeneity gap. However, some limitations still need to be addressed, as observed by [30] and [31]: (i) the model needs a massive amount of labeled data in the pre-training phase; (ii) the environmental cost associated with large-scale training; and (iii) possible model biases caused by large public datasets used for pre-training.

The differences between our work and [81, 73, 70, 30, 31] are : (i) we do not use the Teacher Network with external knowledge or a pre-trained model; and (ii) the multimodal data are compared only once to avoid the model degradation and the heterogeneity gap.

3.2.2 Multimodal FSL without external knowledge

When human beings need to learn a new task never experienced before, they usually try to adapt the learning method, and after a few attempts, they may successfully learn the new task. Similarly, machine learning models can learn from a few data without previous experience or knowledge. In this subsection, we selected models that do not use knowledge transfer from a pre-trained model. Instead, use methods to learn fast from a few data.

Optimization-based learning

The first method is optimization-based learning. Song et al. [32] used Long Short-Term Memory (LSTM) [46] as a meta-learner to optimize the model for the new tasks while keeping the learned knowledge. In this model, data of each modality are extracted separately and represented in the same vector space following decision-level fusion. The disadvantage is that the model must first process the most discriminative data modality to obtain good results. However, this constraint limits the model's applicability. Generally, when the dataset is balanced and consistent, the data are more discriminative, and these facts are independent of its modality.

Graph Neural Network (GNN)

The second method is Graph Neural Network (GNN) [87], a structural pattern recognition. GNN is composed of nodes and edges and can represent complex structures, such as images, texts, and proteins. The nodes and edges representations learned by the graph can be propagated to the adjacent neighbors. Hence, GNN learns the node attributes and topological representation to perform node-level, edge-level, or graph-level prediction tasks. Ding et al. [84] proposed the Graph Prototypical Network combining GNN and Prototypical Network for image classification. The model performs a graph meta-learning for node classification from a few data. In this work, instead of training individual embedding for each node, the model learns a set of node aggregator functions called Graph Convolutional Network (GCN). Next, the prototypes of each class are used to find similarities among nodes.

After extracting image and text data separately, Ji et al. [78] used semantic GNN and visual GNN to propagate text and image information. After the propagation, the decision-level data fusion was used to mitigate the discrepancy between visual and semantic modalities.

Li et al. [79] used CNN to represent image data in the visual space and mapped the semantic word embedding into the same visual space using Graph Convolutional Network (GCN). Then a knowledge transfer mechanism from training to the test phase was applied to perform a task from a few data.

The limitation of applying GNN for FSL is that the model performance decreases as the number of test classes increases. This degradation is because GCN has to predict a wider variety of node classes in a complex topological graph structure, increasing the difficulty of the classification task from a few data.

Generative Adversarial Network (GAN)

The third method is data augmentation, such as Generative Adversarial Network (GAN) [88]. In this method, the data limitation in FSL can be compensated by creating synthetic data in the less represented modality. GAN is composed of a Generator that learns to create synthetic data, and a Discriminator that learns to identify if the data are synthetic or real. Pahde et al. [2] proposed the Multimodal Prototypical Network for image classification. They used GAN to create synthetic images based on the text annotation. The nearest neighbor was used to image clustering and cosine distance to calculate the similarity between the new data and the multimodal prototype. The limitation of this method is that the model needs an initial dataset to create the synthetic data, which may not be available in a sufficient amount for GAN to learn.

Zero-Shot Learning (ZSL)

The fourth method is Zero-Shot Learning (ZSL), in which the model learns from no labeled data of one modality and uses complementary information from other modalities [89]. For instance, the model is trained with image and text to perform the image classification task. Then the model uses only the text annotation to predict the unseen image.

Next, four works using ZSL with feature-level data fusion are described. Islam et al. [75] used Siamese Network to find similarities between audio data and projected these signals into the pre-trained Word2Vec [43] embedding space for semantic data alignment. The Euclidean distance was used for data classification. Li et al. [77] used GAN to learn a bidirectional projection generating images from text, and vice-versa. Bendre et al. [83] combined ZSL with Variational Auto-Encoder (VAE) [90]. VAE is a neural network composed of an encoder and a decoder that learns an optimized data representation. The image and text data were extracted separately and VAE was used to create a shared common space between image and text. Fang et al. [80] used two parallel VAE for image and text to learn the vision-semantic cross-modal alignment. The difference with [83] is that they devised an additional function that helps the model to learn more discriminative representations from multimodal data.

The limitation of this method [75, 77, 83, 80] is that the models are not adapted for FSL relying on large labeled dataset for training.

Next, two works using ZSL and decision-level data fusion are described. Yu et al. [76] proposed an episode based training with ZSL combined with Prototypical Network. The training set was split into subsets to mimic FSL, and the Euclidean distance between the unseen data and the class prototype was used for classification. The second work is from Pan et al. [85] which used VGG-16 [51] – a CNN model with 16 weight layers – to extract the image data into a vector space. Then the text data was projected into the same vector space, and the cosine distance loss was proposed to minimize the alignment distance between image and text.

The before-mentioned works in ZSL show promising results. However, there are some aspects to consider before use in real-world applications: (i) in some methods, the model needs to know the test data in the early phase to train the classifier [77, 83]; (ii) in the scenario where the model has to learn from a few samples, predicting using complementary information of one modality and no data from other modalities will require a more robust set of training data [75, 77, 83, 80].

Other learning methods

Finally, we selected three recent works that use novel methods to identify unseen data. Fan et al. [33] devised an episodic projection scheme to construct a multimodal vector space for FSL outliers detection. First, they projected image data and its correspondent text label into different vector spaces. Then, they removed the common features to construct a unified multimodal vector space adding small perturbations to the samples to mimic the outliers. Cosine and Euclidean distances were used to calculate the similarity between the original data and the outliers. The limitation of this work is that they used a unimodal image dataset. The text data were created from the class labels composed of a few words. The proposed model may find difficulties identifying outliers on perturbations created with real-world text data, such as noise short-texts or long-texts.

Passalis et al. [74] devised a model with the centroid-based loss which uniformly distributes the embedding vectors around the prototype. The model is optimized to learn a minimum distance between prototypes that are used for classification. The model detects image data outliers based on the distance from each centroid.

Munjal et al. [1] proposed a self-supervision method with the Query-Guided Network. Query guidance compares query (seen) data and gallery (unseen) data to perform a task.

The model learns the seen and unseen image data interactions using the Siamese Network and leverages the interaction by re-calibrating the feature maps. These works [33, 1, 74] use special techniques that enhance the model's ability to identify unseen data. Adding perturbation or re-calibrating the data feature map has great potential for further study by extending to multimodal datasets.

3.3 Research Gaps

In the previous section, we presented recent works of multimodal FSL selected by the systematic literature review. In these works, we observed that an efficient embedding function for feature extraction and the multimodal data fusion type are important choices. Depending of these choices, the information of new modalities may overwrite the knowledge already acquired by the algorithm resulting in the model degradation, also known as catastrophic forgetting. This problem was addressed by [32] processing each modality by modality sequentially. However, the model need to process first the most dicriminative modality limiting the applicability.

Several works [81, 78, 2, 76, 85, 74] use the Prototypical Network for the FSL approach suggesting the efficiency of the class prototype representation. However, this method alone as a linear classifier may not help the model generalization [76, 85, 74]. Only [78, 2] used Prototypical Network in combination with other methods to enhance the model generalization and learning capabilities.

The following research gaps were identified through the systematic research review.

- We identified a few works [78, 2] using the Prototypical Network combined with a learnable classifier instead of a linear classifier.
- We did not find publications using the Prototypical Network with an optimization-based meta-learning method.

3.4 Chapter Summary

This chapter describes the details of the systematic literature review related to multimodal FSL. We selected recent state-of-the-art works published after 2019 in this area using a review protocol. The selected works were grouped in two categories: models with external knowledge and without external knowledge.

The first category is represented by models that learn from large datasets (the Teacher Network) and then transfer the learned knowledge to the downstream task (the Student

Network), such as Transformer. The second category is represented by several methods: (i) optimization-based learning; (ii) GNN; (iii) data augmentation; (iv) ZSL; and others.

The selected works were analyzed to identify the key features of multimodal FSL: the modality, the embedding function, the objective function, the loss function, the FSL method, and the data fusion type. The detailed analysis of the selected works enabled the detection of the main advantages of each method, along with the remaining challenges and gaps in multimodal FSL. Among the selected works, GeMGF was most influenced by the concepts of Song et al.[32] that used a meta-learner to optimize the model for the new tasks while keeping the learned knowledge. The main difference with GeMGF are: (i) there is no limitation to process first the most discriminative data modality to obtain good results with GeMGF; (ii) we used Reptile [57] instead of LSTM-based meta-learner; and (iii) we used FSL while Song et al. [32] used the traditional batch-based training. The information gathered from the systematic literature review was used to develop the framework explained in the next chapter.

Chapter 4

Generic Multimodal Gradient-Based Meta Framework

This chapter presents the devised model: the Generic Multimodal Gradient-Based Meta Framework (GeMGF), and details how it addresses the research problems. The framework uses the FSL technique to reduce the lack of massive training data problem. Multimodal learning plays a relevant role in the framework where data from one modality can complement the scarcity of information from other modalities. The framework uses FSL combined with meta-learning to avoid model degradation in real-world scenarios caused by unseen data. Section 4.1 illustrates an overview our framework. Section 4.2 details how the heterogeneous data modalities are represented and aligned, the FSL protocol for data sampling, the Prototype and Relation Networks configurations. Section 4.3 depicts the mechanism of meta-learning. Section 4.4 presents the summary of this chapter.

4.1 Framework Overview

The framework architecture comprises the base learner (Figure 4.1) and the meta-learner (Algorithm 1). The base learner consists of four sub-models: (i) image embedding (sub-model 1); (ii) text embedding (sub-model 2); (iii) multimodal embedding (sub-model 3); (iv) and multimodal FSL (sub-model 4). Image and text data are extracted separately. After the raw data extraction, the model learns the multimodal embedding vector. Next, the Prototypical Network is combined with Relation Network in the multimodal FSL. We train the model end-to-end from scratch with a few samples of data in an episodic way. The model does not use any external knowledge or pre-trained models. The meta-learner is the Reptile-based algorithm detailed in Algorithm 1. It is an optimization-based meta-learning and does not have a separate neural network model. Instead, the meta-learner adjusts the base learner's parameters, helping the overall framework gener-

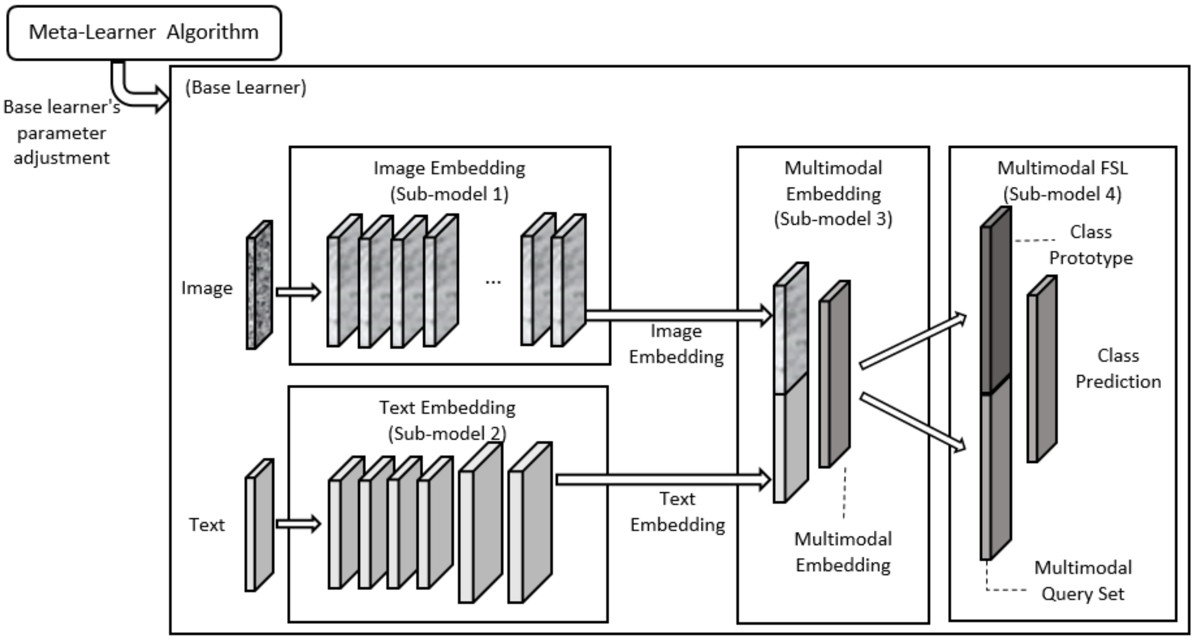

Meta-Learner Algorithm
Base learner's parameter adjustment
(Base Learner)
Image Embedding (Sub-model 1)
Image
...
Image Embedding
Text Embedding (Sub-model 2)
Text
Text Embedding
Multimodal Embedding (Sub-model 3)
Multimodal Embedding
Multimodal FSL (Sub-model 4)
Class Prototype
Class Prediction
Multimodal Query Set

Figure 4.1: Generic Multimodal Gradient-Based Meta Framework

alization. GeMGF architecture is flexible and adaptable to real-world situations where massive training data may not be available.

4.2 Base Learner

The base learner is implemented as one of the two parts of the framework. We will start by describing the multimodal data representation and fusion. In this research we use the modalities of images and texts. Multimodal data usually have different dimensions and structures, making it necessary to reduce the semantic gap among modalities to compare and find similarities [34]. In this process, we can identify two relevant key points: (i) the feature extractor; and (ii) modality alignment choices.

In our framework, the features extractor comprises two sub-models: (i) image embedding (sub-model 1); and (ii) text embedding (sub-model 2). The modality alignment is executed by multimodal embedding (sub-model 3). After the feature extraction and modality alignment, sub-model 4 is responsible for multimodal FSL. The four sub-models are explained in the following subsections.

4.2.1 Image Embedding - Sub-model 1

The image embedding is a modified ResNet [5], as illustrated in Figure 4.2. First, the image raw data is extracted by CNN layer, followed by batch normalization and max pooling layers. We used the pixel size of (180, 180) for all images without data augmentation. Then the model stacks several identity blocks. Each identity block has two or more pairs of CNN layers, followed by a batch normalization layer. The CNN layers calculate the residual information related to the input of each block, while the identity shortcut connections skip the CNN layers without adding extra parameters or calculations. The combination of several stacked identity blocks and identity shortcut connections helps to minimize the effects of vanishing gradient [46] and over-fitting, even for deep neural networks. The original ResNet was presented by He et al. [5] with 50, 101, and 152 identity blocks.

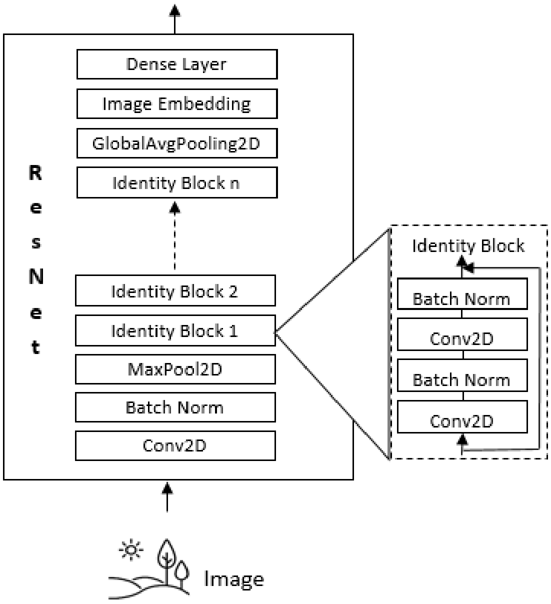


Figure 4.2: Image embedding (sub-model 1).

We chose ResNet because the identity block composition is adaptable to the available computational resource while keeping the data extraction ability. In our framework, we used 30 identity blocks to extract the image embedding, which avoids high resource consumption. Each identity block comprises two pairs of CNN and Batch Normalization layers, as shown in Figure 4.2. Then the model stacks Global Pooling to capture the most relevant information that is used by Image Embedding layer, which is a Dense layer. The last Dense layer is a multi-class classifier. The implementation details of each layers are described in Section 5.2.1 of Chapter 5.

4.2.2 Text Embedding - Sub-model 2

The text data are extracted with Bidirectional Long Short-Term Memory (BiLSTM), as illustrated in Figure 4.3. First the Embedding layer converts words in a numeric vector, followed by BiLSTM. The BiLSTM comprises forward and backward Long Short-Term Memory (LSTM) [46]. LSTM detects relevant features from the input sequence in the early stage and transmits the information over a long distance, thus capturing potential long-term dependencies. BiLSTM helps to capture the context of past and future time steps [48]. The BiLSTM layer is followed by a Time Distributed Layer and a Dense layer, which produces the text embedding. The last Dense layer is a multi-class classifier. We chose BiLSTM because the multimodal datasets (CUB-200-2011 and Oxford-102) used in this work composed of image and text are annotated with long texts. CUB-200-2011 has up to 300 words describing the image data, and in the Oxford-102 dataset, image data are described with up to 120 words (see details in Section 5.1 of Chapter 5). BiLSTM is effective in many application areas [91]. However, this framework can be adapted to short text extraction using CNN or an automatic encoder [3, 92].

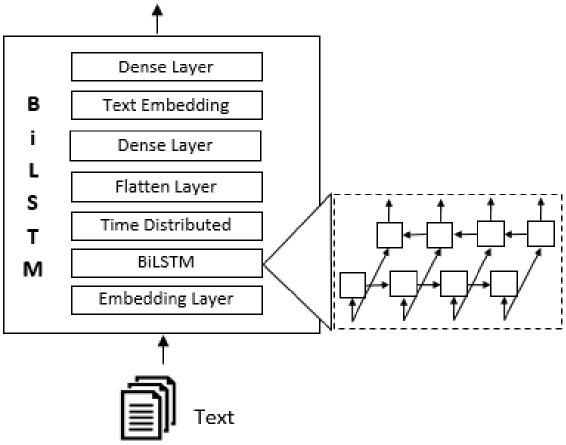


Figure 4.3: Text embedding (sub-model 2).

4.2.3 Multimodal Embedding - Sub-model 3

The second key point when using multimodal data is the modality alignment technique. The human brain learns more efficiently if we process one modality at a time, acquiring knowledge modality by modality [93]. Once the human has learned the knowledge of one modality, the brain can learn the next modality. Inspired by this mechanism, in sequential cross-modal learning [32], data from the first modality are extracted and represented in the vector space S. Then data from the second modality are extracted and represented in S. The semantic gap is reduced in sequential cross-modal learning by mapping each modality data into the same vector space S.

We used a similar mechanism of sequential cross-modal learning [32], extracting one modality at a time. However, instead of representing all modalities in the same vector

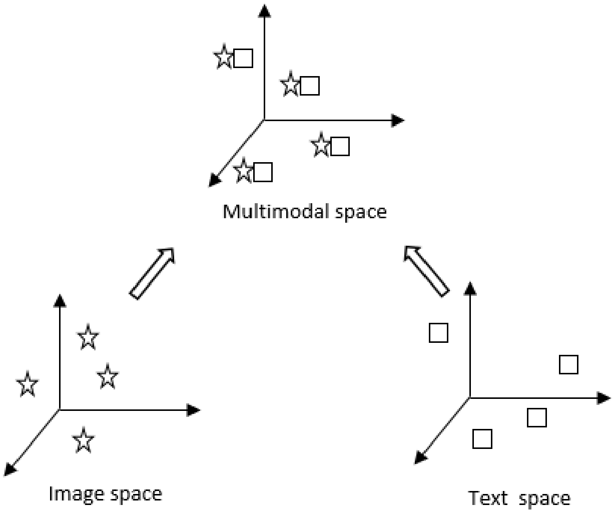


Figure 4.4: Multimodal space

space, initially the image data and text data are represented in a separate vector spaces, as illustrated in Figure 4.4. During the data extraction process, we used the decision-level fusion, in which each modality is individually processed based on an independent decision task. Figure 4.5 illustrates the multimodal data representation and fusion process. First, the sub-model 1 extracts image data using ResNet and learns a multi-class classifier, producing the image embedding vector. Similarly, sub-model 2 extracts text data using BiLSTM and learns another multi-class classifier, producing the text embedding vector.

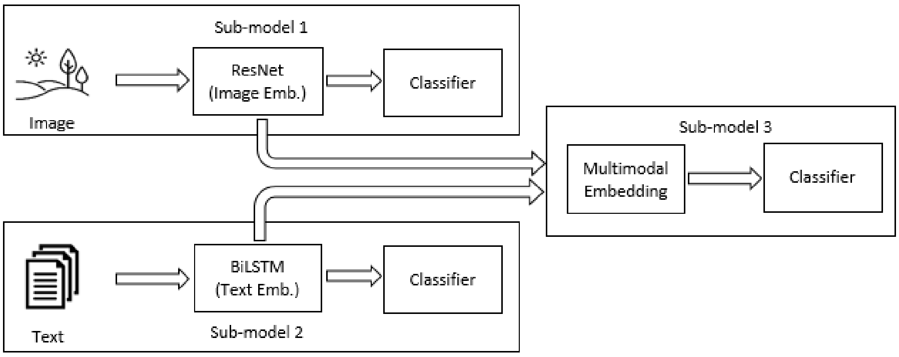


Figure 4.5: Multimodal data representation and fusion process.

After the feature extraction, sub-model 3 concatenates the features from text and image embedding and learns a third multi-class classifier. Equation 4.1 details the concatenation operation, in which i is the $i-th$ element of the dataset, v_i represents a single image and x_i is the annotation text that describes the image v_i. The function f_φ returns the features from BiLSTM embedding, and the function f_ϑ returns the features from

ResNet embedding.

$$Z = concat(f_{\varphi}(x_i), f_{\vartheta}(v_i)) \tag{4.1}$$

Then sub-model 3 learns the alignment of two different modalities by Stochastic Gradient Descent (SGD) and Adam optimizer. The semantic gap is addressed by this last sub-model where image and text data are integrated in the same future space using decision-level fusion. Each sub-model has an independent extractor and decision-making mechanism resulting in a more flexible framework. For example, the extractor of sub-model 2 can be replaced by CNN for short-texts classification.

4.2.4 Multimodal Few-Shot Learning - Sub-model 4

So far, GeMGF has extracted features from the raw data of each modality (sub-model 1 and sub-model 2) and aligned it in a unified vector space (sub-model 3). Now we will use a few samples of the multimodal data and perform a task, such as classification. For this, the multimodal FSL, represented by sub-model 4 in Figure 4.1, is divided into three blocks: (i) Few-Shot data sampling; (ii) Prototype calculation; and (iii) Relation Network.

Few-Shot data sampling

In FSL, the model is trained by episodes, and in each episode, a few samples of data are selected. Following the C-way K-shot notation, C denotes the number of classes and K denotes the number of samples of each class. First, C classes are randomly selected from the entire dataset D, where $D = D_{train} \cup D_{test}$. In this composition D_{train} and D_{test} are disjoint, i.e., $D_{train} \cap D_{test} = \varnothing$. Then K samples of each class are randomly selected from D_{train} to compose the support set $S = \{(v_i,x_i,y_i)\}_{i=1}^{n}$, in which n denotes the number of elements in S, v_i represents a single image, x_i the annotation text that describes v_i, and y_i corresponds to the class label of v_i. In the next step, samples from D_{test} are used to create the disjoint query set $Q = \{(v_j,x_j,y_j)\}_{j=1}^{m}$, where m denotes the number of elements in Q composed by unseen instances of multimodal data.

In each training episode, K labeled samples of each class in the support set S are randomly selected. The sub-model 1 in Figure 4.5 extracts the related features by the image embedding, and sub-model 2 extracts the related features by the text embedding. After a few episodes, the classifier of sub-model 3 learns the multimodal embedding and produces the support set S_m.

In the same way, a few examples of the query set Q are processed to extract the image and text features, creating the multimodal query set Q_m. This FSL data sampling is repeated for each episode to create S_m and Q_m which are used in the following steps.

Prototype Calculation

In the next step, based on the Prototypical Network [6] concepts, we use the multimodal support set S_m to calculate the Class Prototype $ClassP$. Concretely, all data points of each class in S_m are used to calculate the mean vector to represent the Class Prototype. We assume that all data belonging to the same class cluster around $ClassP$ in the feature space. At this point, we can assume that the quality of the class prototype relies on the image extractor (sub-model 1), text extractor (sub-model 2) along with the multimodal data alignment (sub-model 3). Because of this dependence, the multimodal data representation and fusion choices are crucial. By using the decision-level fusion, we can calibrate the sub-models' classifiers separately to avoid over-fitting, which will influence the class prototype. Equation 4.2 details the formula of $ClassP$, where $l = 1, 2..C$. In the Equation 4.2, l identifies the class, C is the maximum number of classes, and S_{m_l} is the multimodal support set of the specific class.

$$ClassP_l = \frac{1}{|S_{m_l}|} \sum S_{m_l} \tag{4.2}$$

After this process, we compare $ClassP$ with unseen multimodal embeddings.

Relation Network

Next, we need to predict the class of elements in the multimodal query set Q_m based on the class prototype learned from the multimodal support set S_m. The model uses Relation Network [7] to learn the relation between the class prototype and the data in Q_m. The goal is to calculate the relation score between these two vectors.

The class prototype $ClassP_l$ and the elements in Q_m denoted by Q_{m_j} are concatenated to learn the score r_{lj}. The model is trained by SGD and uses a binary classifier, where values of r_{lj} close to 1 represents similarity between $ClassP_l$ and Q_{m_j} and 0 represents dissimilarity. The relation function g_ϕ in the Equation 4.3 returns the relation score r_{lj}, where $l = 1, 2..C$. In the Equation 4.3, l identifies the class, C is the maximum number of classes, and $j = 1, 2..m$, where m is the number of elements in the multimodal query set Q_m.

$$r_{lj} = g_\phi(concat(ClassP_l, Q_{m_j})) \tag{4.3}$$

Figure 4.6 illustrates an example of the Relation Network. Lets consider the 5-way 5-shot FSL protocol, which means five classes with five samples per class. $ClassP_1$ is compared with the first element in the multimodal query set data denoted by Q_{m_1}, resulting the score $r_{11} = 0$. Similarly, Q_{m_1} is compared with each class prototype, and the comparison between Class Prototype 5 and Q_{m_1} resulted in the score $r_{51} = 1$, meaning that the model predicted that Q_{m_1} belongs to class 5.

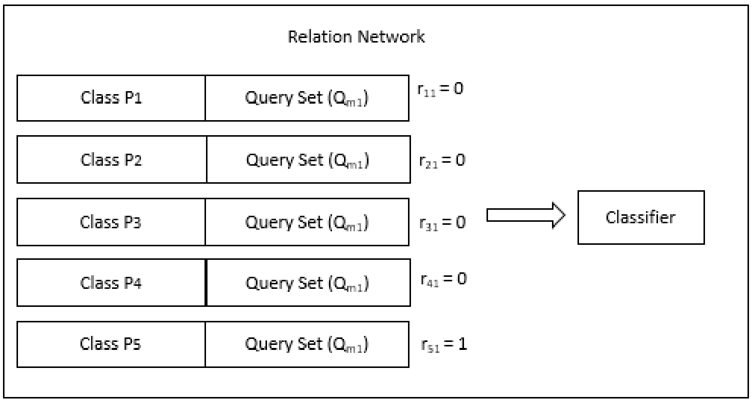


Figure 4.6: Example of Relation Network for 5-way FSL protocol (sub-model 4).

In this way, the Multimodal Few-Shot Learning sub-model learns whether the multimodal query data and the class prototype are from matching categories or not. The advantage of using a relation function instead of a linear classifier (such as Euclidean distance) is that the model can benefit from a learnable non-linear approach rather than fixed metrics [7]. The main difference between our method and the original Relation Network [7] is that we use the class prototype to compare query set data instead of comparing each support set data.

4.3 Multimodal Meta-Learner

The multimodal meta-learner is the second of the two parts that compose the GeMGF framework. It is an optimization-based meta-learning and does not have a separate neural network model. Instead, the meta-learner adjusts the base learner's parameters, helping the overall framework generalization.

FSL methods may struggle to achieve good results since the available training data are severely limited. One possible option is to use external memory, such as a complex pre-trained Transformer model, to transfer the learned knowledge to the FSL model. However, this option limits the applicability to the domain context already known by the pre-trained model. The other option is to use meta-learning jointly with FSL to optimize the entire model's learning capabilities.

In our work, we adopted the second option exploring the flexibility of Reptile [57]: a gradient-based meta-learning approach. The key point of meta-learning is to help the underlying model learn from previous experience, resulting in task generalization. The multimodal meta-learner's details are described in Algorithm 1.

Algorithm 1 Reptile-based Multimodal Meta-Learner

```
 1: Initialize the weights $\phi$
 2: Initialize meta step size $\epsilon$
 3: for  each meta-iteration (outer loop)  do
 4:     for  each episode (inner loop) do
 5:         Construct multimodal support set $S_m$
 6:         Calculate Classs P
 7:         Construct multimodal query set $Q_m$
 8:         Calculate relation score $r_{lj}$
 9:     end for
10:     Calculate $\tilde{\phi} \leftarrow U(\phi)$
11:     Update $\phi \leftarrow \phi + \epsilon(\tilde{\phi} - \phi)$
12:     Adjust $\epsilon$
13: end for
```

First, the weights ϕ are initialized randomly (line 1), and the meta step size ϵ is initialized with a fixed value (line 2). The algorithm runs a few episodes for each meta-iteration to mimic FSL. In the inner loop (lines 4 to 9), for each episode, K samples of C classes are randomly selected, and the corresponding multimodal support set S_m (line 5) and query set Q_m are constructed (line 7). Then, $ClassP$ are calculated (line 6) and used to get the relation score r_{lj} between each Class Prototype and the query set (line 8). In the meta-iteration (outer loop), the new weights $\tilde{\phi}$ are learned by SGD, where U is the SGD operation that updates ϕ (line 10). Next, the weights ϕ are updated moving ϕ closer to the optimal value (line 11). Figure 4.7 illustrates how Reptile works. Let us consider the meta-learner's initial parameter ϕ and a set of tasks $\tau = \{\tau_1, \tau_2, \tau_3\}$. In our case, each task in τ performs a Multimodal FSL, i.e., refers to the inner loop in the Algorithm 1 (lines 4 to 9).

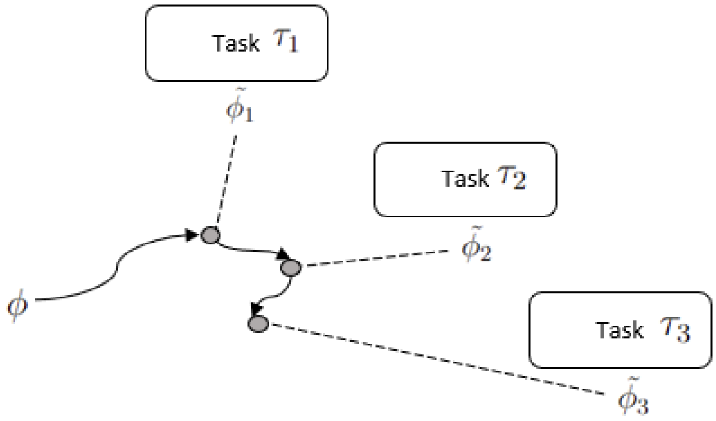


Figure 4.7: Example of Reptile.

Let τ_1 be the first task, the meta-learner performs SGD for a few episodes to get the optimal parameter $\tilde{\phi_1}$. Because the task τ_1 is trained over a few samples of data, the parameter $\tilde{\phi_1}$ is probably over-fitted to τ_1 and cannot be used in the next task. Then, instead of using $\tilde{\phi_1}$, the meta-learner adjusts ϕ slightly closer to $\tilde{\phi_1}$ (line 11 in Algorithm 1) and uses the updated ϕ in the second task τ_2. The optimal parameter $\tilde{\phi_2}$ for τ_2 is learned by SGD, and the parameter ϕ is adjusted in the same way, repeating the processes for all tasks in τ. As a result, ϕ is updated to be closer to $\tilde{\phi}$, where $\tilde{\phi} = \{\tilde{\phi_1}, \tilde{\phi_2}, \tilde{\phi_3}\}$, making ϕ to move in alternate directions, as illustrated by solid lines in Figure 4.7.

By this mechanism, GeMGF learns from a small number of data and keeps the learned knowledge by updating the weight ϕ under the guidance of ϵ. This updating mechanism helps the model keep the knowledge learned in the previous episodes and acquire new knowledge, which ultimately will help the model's generalization.

4.4 Chapter Summary

This chapter details how the Generic Multimodal Gradient-Based Meta Framework addresses the research problems. The framework is divided into the base leaner and the meta-learner. The base learner is responsible for creating the multimodal data, where the modality-specific feature extractor and the modalities alignment choices are relevant. We used a modified ResNet30 for image feature extraction and BiLSTM for text feature extraction, and applied sequential cross-modal learning to extract one modality at a time. Then we used the decision-level fusion, where each modality has an independent decision task. After the feature extraction, the model learns the alignment between image and text data, integrates into the same feature space, and reduces the semantic gap between different modalities.

After the multimodal data creation, FSL is used to train the model in an episodic way, combining Prototypical Network and Relation Network. The Prototypical Network creates a vector representation of each class, and the Relation Network learns the relation among the class prototype and the unseen data. This FSL configuration reduces the dependency on large annotated datasets.

The meta-learner updates the base learner's parameter to help generalization. We use Reptile [57], a gradient-based meta-learning that helps the base learner's training process. In addition to keeping the knowledge learned in previous episodes, the model acquires new knowledge, helping the model generalization.

The growth of computational cost to train complex models is addressed by reducing the number of trainable parameters of the data extractors and avoiding deep networks, which is detailed in Chapter 5 and discussed in Chapter 7.

Chapter 5

Experimental Details

This chapter describes the experimental details of GeMGF. Section 5.1 describes the unimodal and multimodal datasets used to train and evaluate the framework. Section 5.2 describes the implementation details of two variations of GeMGF: the multimodal and the unimodal framework. Section 5.3 presents the summary of this chapter.

5.1 Dataset Description

This section describes the unimodal and multimodal data used in this research. We chose data from different domains, alphabetic and non-alphabetic languages evaluating the framework's adaptability and flexibility. The text data are from the legal area, online newsgroups about a variety of themes, and short texts from Twitter about Ebola epidemic. We used multi-lingual text data with Portuguese, Japanese, and English datasets to analyze the adaptability of our framework to non-alphabetic languages. The image data are related to the medical area: chest x-ray images and blood cell images. The multimodal data are benchmark datasets from the botanical and zoological areas.

The details of all datasets are described in Table 5.1. The column 'Modal' refers to the data modality where image is represented by 'I' and text by 'T'. The column 'Size' is the number of text documents for text data or the number of image files for image data. 'Length' is the maximum number of words or characters of each text data, 'Lang.' refers to the language of text data, 'Total #Class' refers to the total number of classes, 'Avg. per Class' is the average number of samples per class, and the column 'Min. Max' denotes the minimum and maximum samples in the classes. Examples of EN-T, JP-T, Livedoor, DEC6, and 20NG text datasets are presented in Table 5.2 and Table 5.3. Examples of COVID19 and Malaria image datasets are illustrated in Table 5.4. Examples of CUB-200-2011 and Oxford-102 multimodal datasets are illustrated in Table 5.5 and Table 5.6, respectively.

Dataset	**Modal**	**Size**	**Length**	**Lang.**	**Total #Class**	**Avg. p/Class**	**Min. Max.**
EN-T[3]	T	1162	30	EN	5	230	60-531
Tweet50	T	250	30	EN	5	50	50-50
JP-T[3]	T	156	140	JP	4	39	17-74
Livedoor	T	4,572	80	JP	4	870	835-900
DEC6[4]	T	162	150	PT	6	20	10-57
20NG	T	11,314	60	EN	20	500	377-600
COVID19	I	317	-	-	3	105	90-137
Malaria[94]	I	827	-	-	2	413	408-419
CUB-200-2011[95]	T,I	11,788	300	EN	200	60	41-60
Oxford-102[96, 97]	T,I	8,189	120	EN	102	80	40-258

Modal: T for text, I for image data. Size: number of text document for text data or number of image files for image data. Lang.: EN for English dataset, JP for Japanese dataset, PT for Portuguese dataset.

Table 5.1: Description of the datasets used in the experiment.

The following subsections describe each dataset.

5.1.1 Unimodal Dataset

The unimodal data are text datasets used in our previous works [3, 4] related to Natural Language Processing (NLP) and image datasets.

The following text datasets are used in our experiments:

EN-T [3] was collected from Twitter during the 2014/2015 Ebola outbreak. 1162 English tweets were manually annotated into five classes related to the outbreak, including situation reports, economic and social impact, and vaccine development. EN-T contains a class-imbalanced short text with up to 30 words, a minimum of 60 samples and a maximum of 531 samples of tweets in the classes.

Tweet50 was created as a subset of EN-T and used to analyze the results of a class-balanced dataset with a total of 250 tweets equally distributed among the five classes.

JP-T [3] was collected from Twitter during the 2014/2015 Ebola outbreak. JP-T is a small dataset with 156 tweets in Japanese manually annotated into four classes related to the Ebola outbreak. JP-T is a severely class-imbalanced dataset with up to 140 Japanese characters, a minimum of 17 samples and maximum of 74 samples of tweets in the classes.

Dataset	Text examples	Translation
EN-T	Ebola death toll tops 10000.	-
	ebola affecting food security across westafrica study explain t.co/ot4di t.co /ot4di	-
	student remain hesitant attend recent reopen school aft deadly spread	-
JP-T	死者８０００人超に＝西アフリカの #エボラ熱 #WHO	More than 8000 deaths in West Africa #Ebola #WHO
	エボラ出血熱の可能性、世田谷の70代女性-シエラレオネに滞在歴	Possibility of Ebola hemorrhagic fever, a woman in her 70s from Setagaya with a history of staying in Sierra Leone
	シエラレオネ、キューバの医師がエボラに感染	Cuban doctor infected with ebola
Livedoor	東日本大震災から1年、マスコミの災害報道について語る映画が公開	One year after the East Japan earthquake, a movie about the disaster coverage of the mass media is released
	プロがおすすめ、クリスマスにふさわしいオーガニックワイン	Organic wine for the Christmas season suggested by a professional
	Androidを狙うウイルスが急増！スマホに迫る悪質なアプリ 【役立つセキュリティ】	viruses targeting Android has increased! malicious smartphone apps [useful security tips]
DEC6	I) negar provimento ao agravo de instrumento do reclamado; II) conhecer do recurso de revista do reclamado, por contrariedade à Súmula 219, I, do TST, e, no mérito, dar-lhe provimento para excluir da condenação o pagamento dos honorários advocatícios.	I) dismiss the interlocutory appeal of the defendant; II) hear the defendant's appeal for review, contrary to Precedent 219, I, of the TST, and, on the merits, grant it to exclude the payment of attorney fees from the conviction.
	Retirar de pauta - por unanimidade, negar provimento ao agravo e, ante a sua manifesta improcedência, aplicar multa de 2% do valor atualizado da causa, nos termos do art. 1.021, § 4º, do CPC. por solicitação do Excelentíssimo Ministro Augusto César Leite de Carvalho.	Withdraw from the agenda - unanimously dismiss the appeal and, given its rejection manifest, apply a fine of 2% on the updated value of the cause, according to article 1,021, § 4, of the CPC. at the request of the Honorable Minister Augusto César Leite de Carvalho.

Table 5.2: Examples of data from EN-T, JP-T, Livedoor, and DEC6 text datasets.

Livedoor contains 4,572 Japanese news texts collected from Kagle Datasets [1]. It is categorized into four classes related to sports, computers, movies, and shopping. The dataset is class-balanced, with an average of 870 texts per class.

[1] https://www.kaggle.com/datasets/vochicong/livedoor-news

Dataset	Text examples	Translation
20NG	Applied Engineering makes a NuBus card called the QuadraLink which is a board that contains 4 serial ports, which I believe can be used simultaneously. I'm not a user of one of these, but I have installed a couple for people at work (I'm a technician). Hope this helps.	-
	Any lunar satellite needs fuel to do regular orbit corrections, and when its fuel runs out it will crash within months. The orbits of the Apollo motherships changed noticeably during lunar missions lasting only a few days. It is *possible* that there are stable orbits here and there – the Moon's gravitational field is poorly mapped – but we know of none. Perturbations from Sun and Earth are relatively minor issues at low altitudes. The big problem is that the Moon's own gravitational field is quite lumpy due to the irregular distribution of mass within the Moon.	-
	Reasonable doubt dates back to Human Rights. We are now in the time of Civil Rights. Civil Rights are issued by the State with whatever strings attached they choose as the Grantor of said rights. And if that means that verdicts are determined by the needs of the state rather than by guilt or innocence in a traditional sense, so be it. Being subjective rather than objective may make it harder to anticipate what is right, and you may be sacrificed for being wrong inadvertently once in a while, but that really is a small price to pay for the common good don't you think?	-

Table 5.3: Examples of data from 20NG text dataset.

DEC6 [4] is composed of 162 movements text in Portuguese extracted from the Brazilian Superior Labour Court. Movements reflect the significant updates and phases of the judicial case life-cycle and may contain the text of the judge's analysis and decisions. The case's final decision is published based on this text. From an average of 30 decision types, the top six most used types are used in this dataset. Some examples of decision type of Superior Labour Court are: appeal granted, trial postponed, and the application denied. These movement texts were categorized by legal experts into six decision types and used as multi-class categorization. DEC6 is a class-imbalanced long text dataset with a minimum of 10 samples and maximum of 57 samples of data in the classes.

20NG is a multi-class benchmark text dataset collected from twenty on-line newsgroups and extracted from Scikit-learn datasets [2]. 20NG is a large dataset containing long texts with average of 500 documents per class. It contains 20,000 documents in English categorized into 20 classes. We used a subset of 11,314 documents and the maximum of 60 words of each document.

We chose two image datasets from the medical area, as follow:

COVID19 contains 317 images of chest x-rays downloaded from Kaggle Datasets [3]. It has 137 chest images of COVID-19, 90 normal chest images, and 90 chest images of viral pneumonia.

Malaria [94] is available at Tensorflow Datasets [4]. The Malaria dataset contains a total of 27,558 segmented cell images from the thin blood smear slide categorized into two classes: parasitized cells and uninfected cells. A subset of 827 images were used in this work. It is a class-balanced dataset with average of 413 images per class.

5.1.2 Multimodal Dataset

Two benchmark multimodal datasets containing images and long texts are used in our work: CUB-200-2011 and Oxford-102. The first is class-balanced, and the second is a class-imbalanced dataset.

CUB-200-2011 [95] is a publicly available multimodal dataset [5], which contains 11,788 images of 200 species of birds. Each specie represents a category or class, and there are, in average, 60 birds samples for each class. All images are annotated with up to 312 English text attributes related to color of the bird, pattern and shape of a specific part. CUB-200-2011 is widely used in the multimodal research [2, 81, 76, 83]

Oxford-102 [96] contains 8189 images of flowers belonging to 102 different categories commonly occurring in the United Kingdom. This dataset [6] has, in average, 80 images per class, with minimum of 40 and maximum of 258 images. Each image is annotated with ten English textual descriptions provided by [97].

[2] https://scikit-learn.org

[3] https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset

[4] https://www.tensorflow.org/datasets/catalog/malaria

[5] http://www.vision.caltech.edu/datasets/cub_200_2011/

[6] https://www.robots.ox.ac.uk/ṽgg/data/flowers/102/

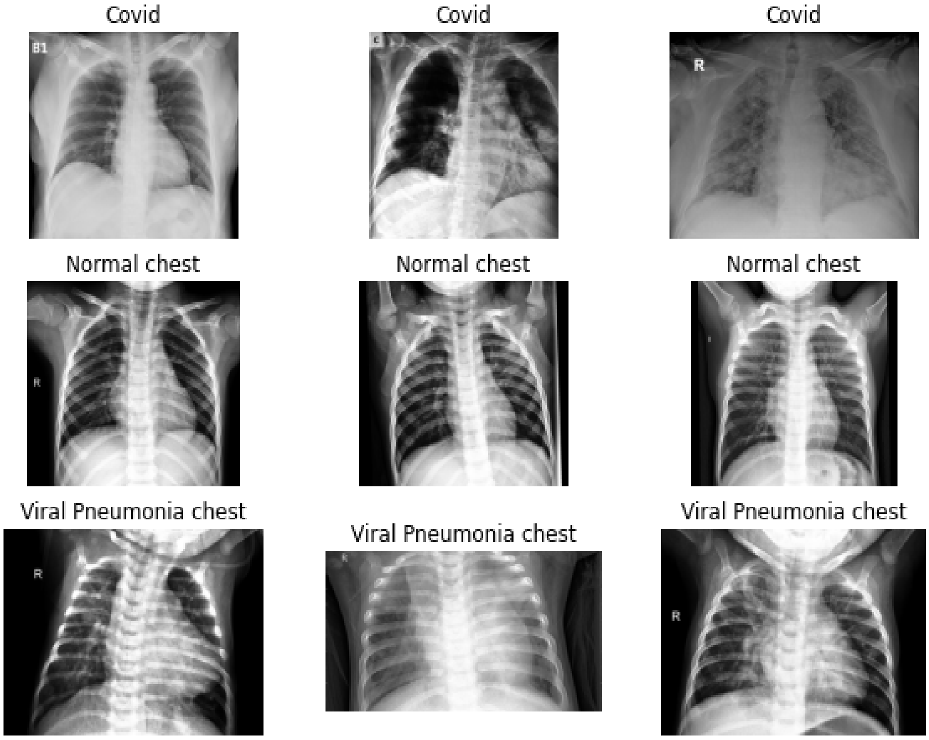
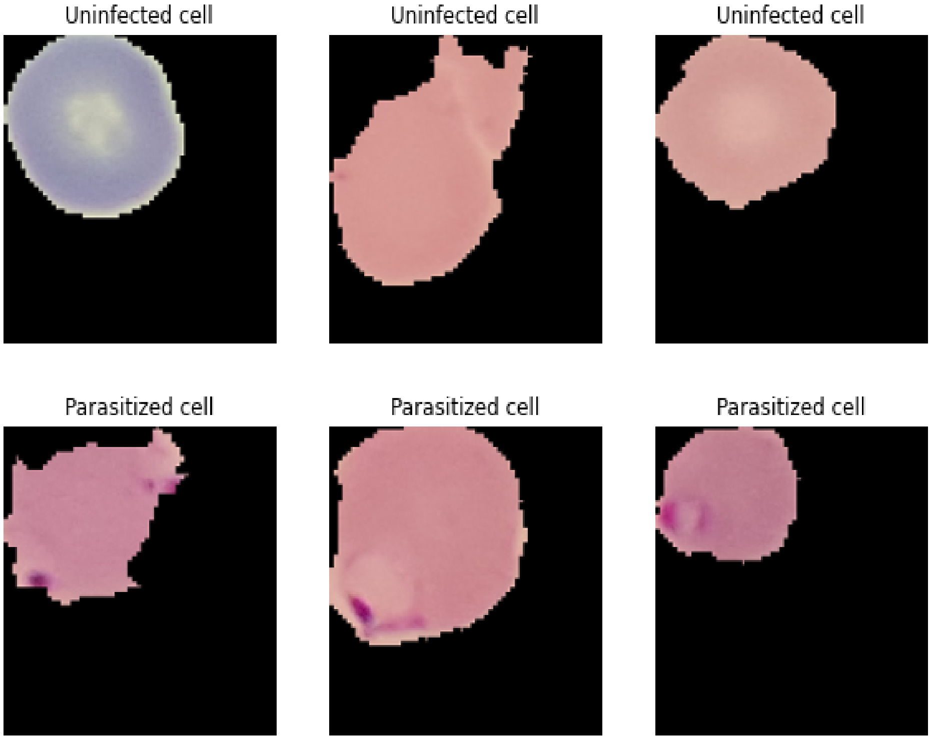
Dataset	Image examples
COVID19	 Covid Covid Covid Normal chest Normal chest Normal chest Viral Pneumonia chest Viral Pneumonia chest Viral Pneumonia chest
Malaria	 Uninfected cell Uninfected cell Uninfected cell Parasitized cell Parasitized cell Parasitized cell

Table 5.4: Examples of COVID19 and Malaria image datasets.

Image with text annotation	
wing pattern solid, crown color brown, bill color black, leg color buff, primary color brown, belly pattern solid, tail pattern solid, back pattern solid, shape upright-perching water-like, size medium (9 - 16 in), wing shape pointed-wings, belly color brown, nape color brown, under tail color brown, forehead color brown, bill length about the same as head, eye color brown, throat color brown, breast color brown, head pattern unique pattern, upper tail color brown, tail shape squared tail, back color brown, breast pattern solid, underparts color grey, underparts color brown, upper parts color brown, wing color brown, bill shape hooked seabird	
crown color brown, bill color buff, bill color black, bill color grey, leg color black, primary color buff, primary color yellow, primary color grey, primary color brown, belly pattern solid, shape perching-like, size very small (3 - 5 in), wing shape rounded-wings, belly color yellow, belly color grey, nape color buff, nape color grey, nape color brown, forehead color brown, bill length shorter than head, eye color black, throat color buff, throat color grey, breast color yellow, breast color grey, head pattern plain, breast pattern solid, underparts color yellow, underparts color grey, wing color brown, bill shape all-purpose	
wing pattern solid, crown color grey, bill color grey, leg color buff, primary color yellow, primary color grey, belly pattern solid, tail pattern solid, back pattern solid, shape hummingbird-like, size very small (3 - 5 in), wing shape rounded-wings, belly color yellow, nape color grey, under tail color grey, forehead color grey, bill length shorter than head, eye color black, throat color yellow, breast color yellow, head pattern capped, upper tail color grey, tail shape notched tail, back color grey, breast pattern solid, underparts color yellow, upper parts color grey, wing color grey, bill shape all-purpose	

Table 5.5: Examples of image and text annotation from CUB-200-2011 dataset.

5.2 Implementation Details

This section describes the implementation details of two variations of GeMGF: the multimodal and the unimodal framework. The former is the original version of our framework, and the latter is the unimodal version used in single-modal domain problems. Subsection 5.2.1 describes the base learner's implementation of the multimodal framework. Subsection 5.2.2 describes the base learner's implementation of the unimodal framework. Subsection 5.2.3 describes the tools and libraries used in this work. Subsection 5.2.4 details

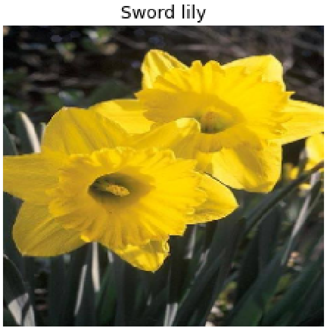
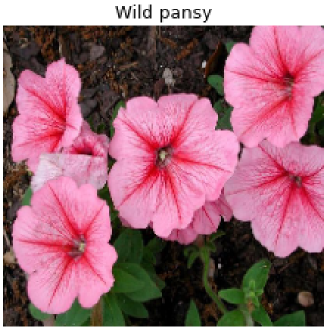
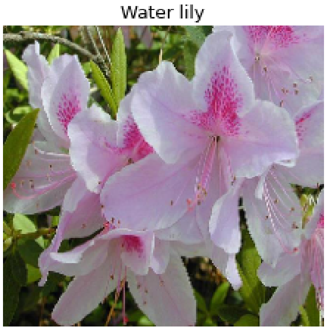
Image with text annotation	
the outer petals are oval in shape,inner petals are rounded and yellow in color two flowers with yellow petals and yellow pistils protruding out the middle. the pedals of this flower are yellow with a long stigma this flower is yellow in color, with petals that are ruffled. the flower has a bright yellow colored petals and even its stamen are of the same color	
this flower is pink in color, with petals that have dark veins. this flower has petals that are pink with purple lines the petals on this flower are pink with red veins. this flower has large pink petals and bright pink stripes going down the middle of them the pink flower has petals that are soft, smooth, thin and enclosing stamen that has white anthers	
this is a white flower with many petals that have pink areas on them. the petals have curled edges and pink details with white filaments. this flower is white and pink in color, with petals that are spotted. this flower has petals that are white with pink dots these beautiful flowers is pink and white in color with long stamen	

Table 5.6: Examples of image and text annotation from Oxford-102 dataset.

the hyperparameters used in the multimodal and unimodal frameworks.

5.2.1 Multimodal Framework Details

Figure 5.1 illustrates the base learner's layers details with four sub-models: (i) image embedding (sub-model 1); (ii) text embedding (sub-model 2); (iii) multimodal embedding (sub-model 3); and (iv) multimodal FSL (sub-model 4). The dimensions of each layer in the illustration assume the 5-way 5-shot FSL protocol, i.e., five classes and five samples per class.

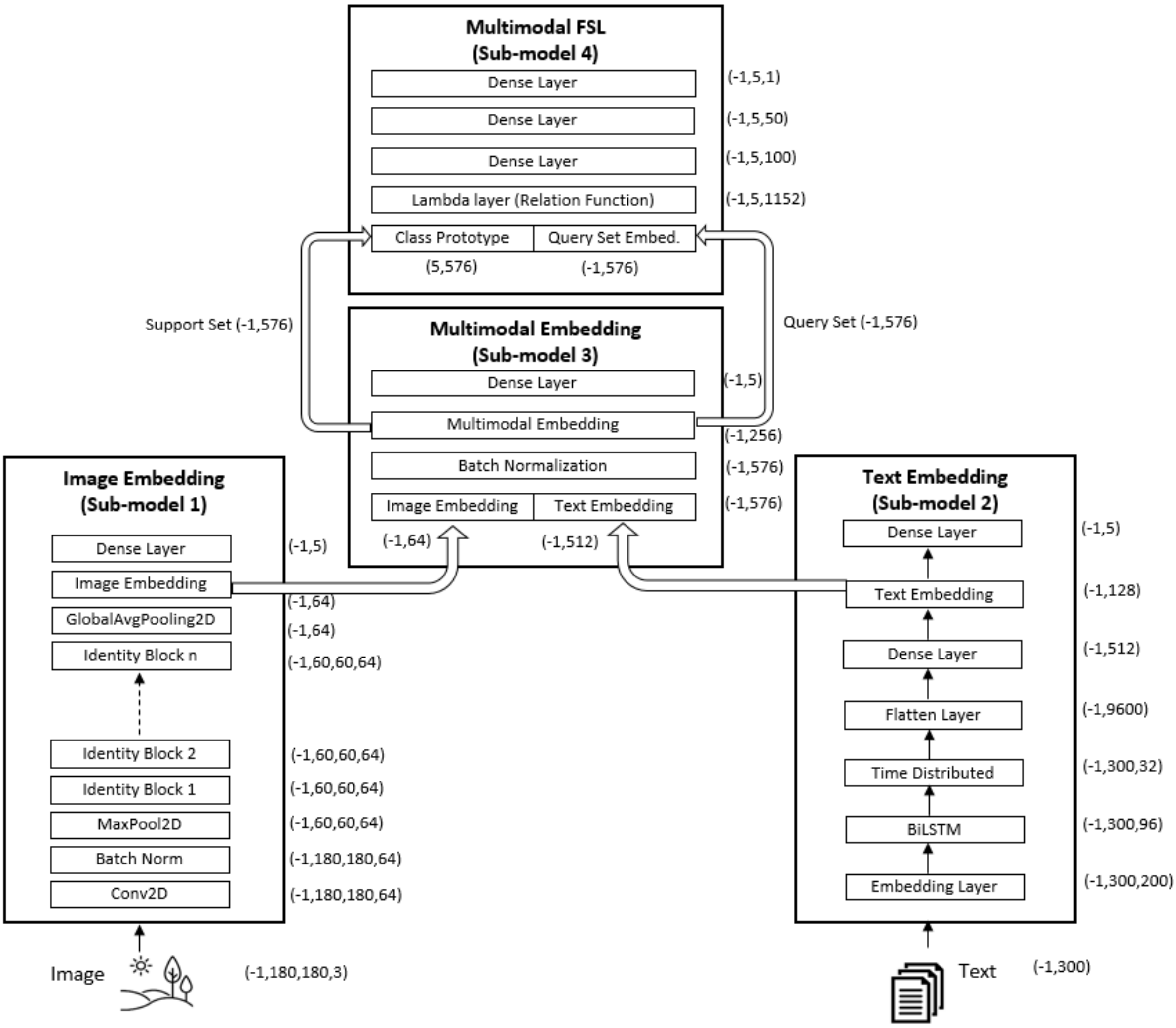


Figure 5.1: Multimodal Framework implementation details.

Image Embedding - Sub-model 1

The input layer has the shape (-1, 180, 180, 3) where -1 represents the variable batch size, 180 refers to the pixel size, and three refers to the RGB color channels. The input layer is followed by the 'Conv2D' layer, which is a two-dimensional CNN. Then the 'Conv2D' is followed by the 'Batch Norm' layer, which is a batch normalization layer responsible for normalizing the input of each layer for every mini-batch. The 'Conv2D' and the 'Batch Norm' layers have the same shape (-1, 180, 180, 64), where 64 refers to the number of filters. The 'MaxPool2D' is a two-dimensional max pooling layer that considers the most relevant features and reduces the dimension to (-1, 60, 60, 64). Then 30 Identity Blocks are stacked, followed by the 'GlobalAvgPooling2D' layer, a global average pooling layer that reduces the dimension to (-1, 64). Next, the image embedding layer keeps the same shape, and the last dense layer is activated by the Softmax function to classify the image embedding into one of the five classes. The image embedding layer is a customized dense layer responsible for providing the embedding vector used in the multimodal embedding (sub-model 3).

Text Embedding - Sub-model 2

In this sub-model, the input layer has the shape (-1, 300), where 300 is the maximum number of words in one text annotation. The embedding layer has the shape (-1, 300, 200), where 200 refers to the size of the word embedding vector learned from scratch during training. The following BiLSTM layer has the shape (-1, 300, 96), where 96 is the sum of the internal forward and backward LSTM layers with 48 neurons each. Next, the time-distributed layer with the shape (-1, 300, 32) applies the same weights to the output of the previous layer for one time step at a time. The following flatten and dense layers reduce the dimension to (-1, 512). Next, the text embedding layer is a customized dense layer that takes the output of the previous layer and produces the text embedding vector used in the multimodal embedding (sub-model 3). The last dense layer is a multi-class classifier that categorizes the input text into one of five classes activated by the Softmax function.

Multimodal Embedding - Sub-model 3

After the image and text raw data extractions, the model learns the multimodal embedding vector. The image and text vectors obtained from the previous Image Embedding (sub-model 1) and Text Embedding (sub-model 2) are concatenated and processed by the batch normalization layer of the shape (-1, 576). Then the multimodal embedding layer, which is a customized dense layer, takes the output of the previous layer and produces the multimodal vector. As with Text and Image Embedding sub-models, the last dense layer classifies the multimodal embedding into one of the five classes activated by the Softmax function. The key point of this sub-model is that multimodal embedding extracted from the support set is used in sub-model 4 to calculate the class prototype. On the other hand, the multimodal embedding extracted from the query set is used in sub-model 4 for testing.

Multimodal FSL - Sub-model 4

Next, the Prototypical Network is combined with the Relation Network in the multimodal FSL sub-model. First, the class prototype calculated from the support set and the query set embeddings are concatenated. Then the lambda layer is used to call an arbitrary expression as a layer. The lambda layer of the shape (-1, 5, 1152) calls the Relation function and calculates the relation score between the class prototype and the query set embedding. In the next step, the sub-model stacks three dense layers to extract the most relevant neuron in the last dense layer, which is a binary classifier activated by the Sigmoid function.

This subsection described the implementation details of the four sub-models that compose the Multimodal Framework. The next subsection details the Unimodal Framework.

5.2.2 Unimodal Framework Details

The GeMGF architecture can be adapted to unimodal data. First, we assume that the unimodal data is a short text. Similarly to the multimodal framework, it comprises the meta-learner and the base learner. Since there is only one modality, the multimodal embedding sub-model is not required. Thus, the base learner is simpler than the multimodal framework, having only two sub-models: (i) unimodal embedding (sub-model 1) and (ii) Few-Shot Learning (sub-model 2). Figure 5.2 illustrates an example of this framework using short text. The dimensions of each layer in the illustration assume the 5-way 5-shot FSL protocol.

Text Embedding - Sub-model 1

The input layer has the shape (-1, 38), where 38 is the maximum number of words in the short text, followed by the embedding layer. This layer has the shape (-1, 38, 300), where we use the word embedding vector learned from scratch with the dimension of 300. The text extractor used in this illustration is a triplet parallel 'Conv2D' with 32 filters followed by the 'MaxPool2D' layer. The same composition of the 'Conv2D' and the 'MaxPool2D' layers is repeated and the output is converted by the flatten layer into the shape (-1, 2400). The text embedding layer takes the output of the previous layer and produces the text embedding vector used in the Few-Shot Learning (sub-model 2). The last dense layer classifies the text embedding learned from CNN into one of the five classes using the Softmax activation function.

Few-Shot Learning - Sub-model 2

This sub-model has the same mechanism and layers of Multimodal FSL. We only need to adjust the class prototype shape to (5,2400) and the query set embedding shape to (-1,2400) according to the text embedding size extracted from the previous sub-model.

In this subsection, we illustrated the implementation details of the unimodal framework for short text domain problems using CNN. However, the flexibility of GeMGF enables the CNN to be replaced by BiLSTM for long text.

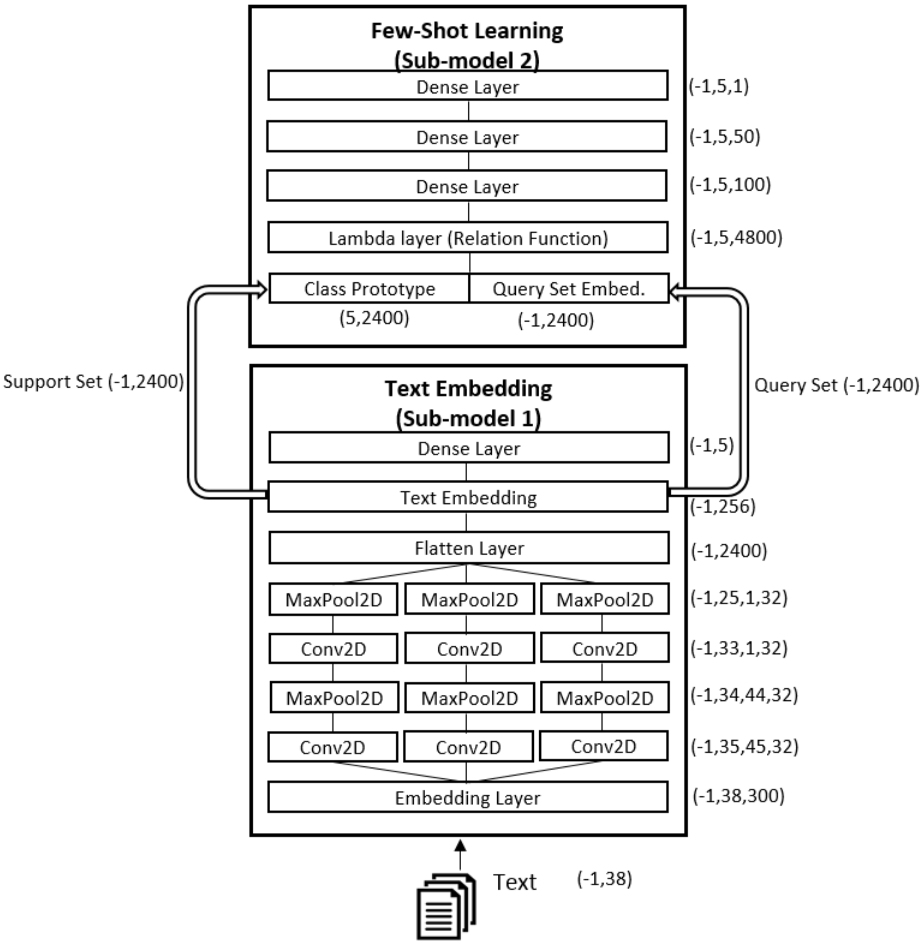


Figure 5.2: Unimodal Framework implementation details.

Adaptation of the Unimodal Framework for Image

The unimodal framework can be adapted to process image data. The framework comprises the meta-learner and the base learner, in which the base learner has two sub-models, as illustrated in Figure 5.3. The text embedding (sub-model 1) is replaced by the image embedding. The dimensions of each layer in the illustration assume the 5-way 5-shot FSL protocol. The image embedding (sub-model 1) has the same ResNet architecture used in the multimodal framework. In the sub-model 2, the class prototype shape is adjusted to (5,64) and the query set embedding shape to (-1,64), according to the image embedding shape extracted from the sub-model 1.

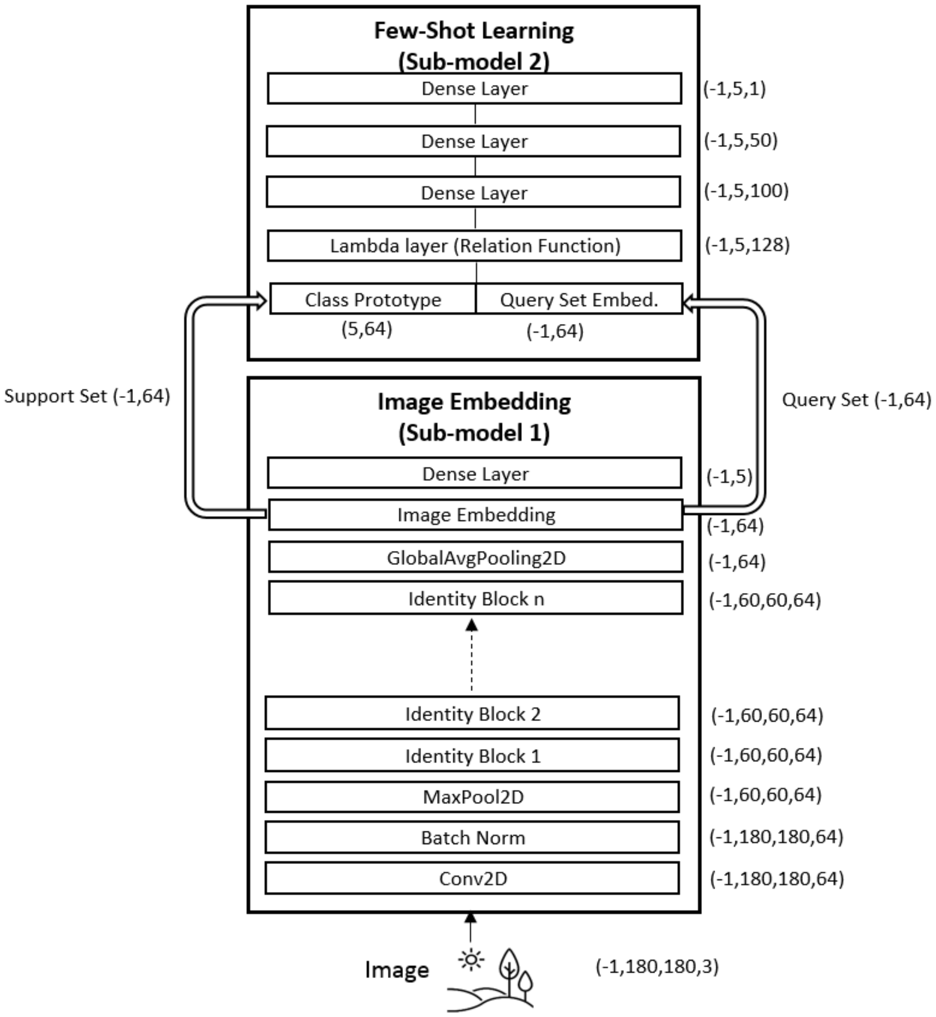


Figure 5.3: Unimodal framework for image implementation details.

5.2.3 Tools and Libraries

We used the following open-source tools to develop GeMGF: (i) Tensorflow[7] 2.10.0 was used to create customized models and layers; (ii) Keras[8] 2.8.0 for image extraction (modified ResNet), text extraction (BiLSTM, CNN) and vocabulary creation; (iii) Scikit-learn[9] 1.0.2 for metric computation, random data split; (iv) Python[10] 3.7.13, Keras-flops[11] for FLOP measurement; Pandas[12] 1.3.5; and Numpy[13] 1.21.6 were used to manipulate and transform data. All the training procedures were run on Google Colab[14] free platform and

[7] https://www.tensorflow.org/

[8] https://keras.io/

[9] https://scikit-learn.org/

[10] https://www.python.org/

[11] https://pypi.org/project/keras-flops/

[12] https://pandas.pydata.org/

[13] https://numpy.org/

[14] https://colab.research.google.com/

to avoid extra resource consumption, we used the following setup: (i) Intel(R) Xeon(R) CPU @ 2.30GHz to read raw image files; (ii) and GPU Tesla T4 for all the remaining procedures.

5.2.4 Hyperparameters

All datasets used in the experiments were split into two disjoint subsets to train our model: $D_{train} \cap D_{test} = \varnothing$. The split ratio was 70% for D_{train} and 30% for D_{test}, with both subsets composed of samples of all classes, but in a disjoint way. Then D_{train} was used for the support set and D_{test} for the query set.

The standard FSL training protocol adopted was 5-way 5-shots, randomly selecting five classes out of the total of classes with five samples each. This FSL training was repeated ten times, randomly selecting five classes for each repetition. Other FSL protocols such as 6-way 1-shot for the DEC6 dataset was adopted depending on the class distribution and number of samples available per class.

Table 5.7 describes the hyperparameters used in the multimodal GeMGF. The first column refers to the sub-model, and the second column refers to the loss function used in each sub-model, in which SCCE stands for Sparse Categorical Cross-Entropy, and MSE stands for Mean Square Error. The third and fourth columns refer to the optimizer and the respective learning rates. The column 'Parms.'refers to the number of trainable parameters. For each meta iteration, 5 episodes were run as FSL procedure. The column 'Meta Iter.'refers to the maximum meta iterations that were dynamically adjusted with early stopping if the results do not improve after 8 meta iterations, i.e., patience set to 8. Finally, the last column refers to the meta step size initial value that was adjusted during the training.

The hyperparameter combination kept GeMGF simple and compact, with 14 million parameters in the multimodal version, as detailed in Table 5.7.

Sub-model	**Loss**	**Optimizer**	**Learn. Rate**	**Parms.**	**Episodes**	**Meta Iter.**	**Patience**	**Meta Step Size**
Image Embed.	SCCE	Adam	0.003	3.0 M	5	-	-	-
Text Embed.	SCCE	Adam	0.001	10.2 M	5	-	-	-
MultiM. Embed.	SCCE	Adam	0.003	302 K	5	-	-	-
MultiM. FSL	MSE	Adam	0.003	120 K	5	-	-	-
Meta-Learning	-	-	-	-	-	100	8	0.25

SCCE (Sparse Categorical Cross-Entropy), MSE (Mean Square Error), M (million), K (thousand).

Table 5.7: Hyperparameters settings of multimodal GeMGF.

The hyperparameters used in the unimodal GeMGF for text data are detailed in 5.8. We used two text embedding architectures: CNN2L for short texts and BiLSTM for long texts. The evaluation was based on six text datasets, with different features such as sizes, text length, and distribution among classes. Each of these features influenced the hyperparameter choices. The model trainable parameters number was influenced by the dataset vocabulary size, the embedding layer size, and the text length.

Dataset	**Text** Embbed.	**Opti-** **mizer**	**Learn.** **Rate**	**Parms.**	**Epi-** **sodes**	**Meta** **Iter.**	**Patience**	**Meta** **Step Size**
EN-T	CNN2L	Adam	0.001	4.0 M	5	200	-	0.25
Tweet250	CNN2L	Adam	0.001	3.0 M	5	100	8	0.25
JP-T	CNN2L	Adam	0.001	10.5 M	5	200	8	0.25
Livedoor	BiLSTM	Adam	0.001	3.7 M	5	200	8	0.25
DEC6	BiLSTM	Adam	0.001	5.7 M	5	100	8	0.25
20NG	BiLSTM	Adam	0.001	15.5 M	5	200	8	0.25
CUB-200-2011	BiLSTM	Adam	0.003	10.4 M	5	100	8	0.25
Oxford-102	BiLSTM	Adam	0.003	4.9 M	5	200	8	0.25

M (million).

Table 5.8: Hyperparameters settings of unimodal GeMGF for text data.

Each word is represented as a numeric value to perform the text categorization. In the alphabetic language (Portuguese and English), we adopted the word-level approach using the space as a word separator. However, Japanese text usually does not have space separations between words, as illustrated in the examples of Table 5.2. Unlike alphabetic languages, there is no clear word boundary for Chinese, Japanese, and Korean texts making it difficult to apply language processing methods that assume words as the basic construct. Therefore, the character-level approach was used for the Japanese dataset to produce better results [98]. For this reason, the unimodal framework for JP-T dataset with character-level approach has 10.5 million parameters while the model for EN-T dataset with word-level approach has 4 million parameters.

Table 5.9 details the hyperparameters used in the unimodal framework for image. The ResNet30 was used to extract the image embedding of all datasets. The column 'Pixels' refers to the size (high, width) of the image. The framework used 4.5 million parameters for all datasets. The learning rate for image data was set to 0.003 for all datasets.

Dataset	**Image** Embbed.	**Opti-** **mizer**	**Pixels**	**Parms.**	**Epi-** **sodes**	**Meta** **Iter.**	**Patience**	**Meta** **Step Size**
COVID19	ResNet30	Adam	(224,224)	4.5 M	5	200	8	0.25
Malaria	ResNet30	Adam	(180,180)	4.5 M	5	300	8	0.25
CUB-200-2011	ResNet30	Adam	(180,180)	4.5 M	5	300	8	0.25
Oxford-102	ResNet30	Adam	(180,180)	4.5 M	5	200	8	0.25

M (million).

Table 5.9: Hyperparameters settings of unimodal GeMGF for image data.

5.3 Chapter Summary

This chapter describes the experimental details of GeMGF. The evaluation of our framework was conducted using ten datasets from different domains and characteristics. The text data are from the legal area, online news groups, short and long texts, and alphabetic and non-alphabetic languages. The image data are related to the medical domain and botanical and zoological areas.

The implementation details of two variations of GeMGF are described: the multimodal and the unimodal framework. For each variation, we described the implementation details such as the dimension of each layer and the activation function used.

The tools and libraries used to develop GeMGF are detailed along with the hyperparameter setting for the multimodal and unimodal framework.

Chapter 6

Performance Evaluation

This chapter describes the experiment results of the unimodal GeMGF with two data compositions: text data only and image data only. Then the evaluation results of the multimodal GeMGF for image and text data are described. The purpose of this data split is to analyze the behavior of GeMGF for each situation described in the following sections: Section 6.1 details the results for the unimodal framework; Section 6.2 describes the results of the multimodal framework; Section 6.3 compares GeMGF with baseline models and state-of-the-art architectures; and Section 6.4 presents the summary of this chapter.

6.1 Results of Unimodal Framework

This Section details the evaluation of the unimodal framework using ten datasets: six text datasets, two image datasets, and two multimodal datasets. Through the experiments, we analyze the framework's dependency on the data quantity, quality, the data distribution between classes, and the text data languages.

6.1.1 Results of Unimodal Framework for Text

In this Subsection, we present the experiment results of the unimodal version of GeMGF using only text data. This experiment aims to analyze the framework adaptability to two types of text embedding sub-models: CNN2L for short text and BiLSTM for long text. We used five real-world text datasets (EN-T, Tweet250, JP-T, Livedoor, and DEC6) to evaluate our framework with heterogeneous and challenging scenarios: (i) noisy short texts, (ii) legal domain long text, and (iii) multi-lingual texts. We also used three widely adopted benchmark datasets (20NG, Oxford-102, and CUB-200-2011).

The results are detailed in Table 6.1. The column 'Dataset' refers to the name of the dataset, 'FSL' is the data sampling protocol in which 6-way 1-shot means six classes and one sample per class. The column 'Embed.' refers to the data embedding method, 'Meta Iter.' is the number of training iterations, and the following three columns are the evaluation metrics: mean accuracy, precision, and F1-score, at a 95% of confidence interval.

Dataset	**FSL**	**Embed.**	**Meta Iter.**	**Accuracy(%)**	**Precision(%)**	**F1-score(%)**
EN-T	5-way 20-shot	CNN2L	200	77.20 ± 0.12	77.49 ± 0.16	76.06 ± 0.13
Tweet250	5-way 10-shot	CNN2L	100	89.20 ± 0.08	91.10 ± 0.06	89.10 ± 0.08
JP-T	4-way 4-shot	CNN2L	200	94.98 ± 0.02	95.99 ± 0.02	94.92 ± 0.02
Livedoor	4-way 4-shot	BiLSTM	200	93.75 ± 0.01	95.00 ± 0.01	93.65 ± 0.01
DEC6	6-way 1-shot	BiLSTM	100	95.00 ± 0.04	96.67 ± 0.03	94.67 ± 0.04
20NG	5-way 5-shot	BiLSTM	200	74.40 ± 0.05	74.85 ± 0.08	73.32 ± 0.07
CUB-200-2011	5-way 5-shot	BiLSTM	100	93.20 ± 0.03	94.10 ± 0.02	93.10 ± 0.03
Oxford-102	5-way 5-shot	BiLSTM	200	95.60 ± 0.04	96.20 ± 0.03	95.55 ± 0.04

Results at a 95% of confidence interval (average accuracy, precision and F1-score ± standard deviation).

Table 6.1: Experiment results of unimodal GeMGF using only text data.

First, we analyzed the results of our framework using EN-T and Tweet250 datasets. For both datasets composed of English short messages from Twitter, we used a model comprised of two layers of CNN in the text embedding sub-model, illustrated in Figure 5.2 of Chapter 5. GeMGF performed poorly with EN-T using 5 or 10 samples per class and had difficulties learning from short and noisy tweets with a class-imbalanced distribution. The best result for EN-T dataset was using 20 samples of each class (5-way 20-shot) trained over 200 meta iterations achieving the average F1-score of 76.06%. On the other hand, the model performed better with Tweet250, the class-balanced version of EN-T, achieving an average F1-score of 89.10% with less training (100 meta iterations) and less training data (5-way 10-shot). These results suggest that for a small dataset composed of noise data, the FSL method may need a class-balanced data distribution to achieve good results.

Next, we analyzed the results of JP-T, a small Japanese dataset with 156 texts with an average of 39 texts per class. The unimodal GeMGF framework was adjusted to process non-alphabetic languages by using character-level text extraction rather than word-level before being processed by CNN2L. GeMGF could adapt well to the small dataset achieving the average F1-score of 94.92%, even without the knowledge of pre-trained word embedding. Then we evaluated Livedoor, a large Japanese class-balanced dataset. Our framework achieved F1-score of 93.65% to classify the text data into four categories with four samples per class (4-way 4-shot).

For the following four datasets composed of long texts, the text embedding sub-model was changed to BiLSTM to explore the flexibility and adaptability of GeMGF with different embedding models. DEC6 is a small Portuguese dataset related to legal domain with only 162 texts with minimum of 10 samples for class, and because of this data restriction, we used 6-way 1-shot, i.e, only one sample per class. Despite the limited number of data for each class, GeMGF achieved 94.67% of average F1-score trained over 100 meta iterations. This good result is probably because DEC6 is composed of long text that may contain more useful and clean information than short noisy messages used in the EN-T dataset.

The second long text dataset is 20NG: the largest in our experiments with a total of 11,314 English texts, an average of 500 samples per class distributed over 20 categories. The results for 5-way 5-shot was 73.32% of F1-score, worse than DEC6, which is also a long text. The complexity of NLP models in terms of the number of parameters depends on the text length, the vocabulary size, and the dataset size. For the model simplicity, the text length of 20NG was decreased from 1200 words to 60 words. The results of 73.32% suggest that useful information may have lost in this process.

CUB-200-2011 is a class-balanced English dataset with 11,788 data, where five classes were randomly selected from 200. The model was trained over 100 meta iterations with 5-way 5-shot composition with a maximum of 300 words achieving 93.10% of F1-score. Oxford-102 is a class-imbalanced English dataset with 8,189 clean textual annotations describing each flower. The model was trained over 200 meta iterations with 5-way 5-shot, where five classes were randomly selected from 102 achieving 95.55% of F1-score.

6.1.2 Results of Unimodal Framework for Image

This Subsection describes the experiment results of the unimodal version of GeMGF using only image data. To evaluate the framework in different domains, we used two benchmark datasets (CUB-200-2011 and Oxford-102) and two datasets from a medical domain (COVID19 and Malaria). The image feature was extracted by ResNet30, as illustrated in Figure 5.3 of Chapter 5.

The results are described in Table 6.2. The first column represents the dataset, 'FSL' describes the data composition where 5-way 10-shot means five classes with ten samples each. The column 'Embed.' represents the embedding model used for data extraction, 'Meta Iter.' is the meta iteration, and the last tree columns are the metrics: mean accuracy, precision, and F1-score with the correspondent standard deviation. The experiment results are at a 95% of confidence interval.

First, we analyzed the two benchmark datasets. The 5-way 10-shot protocol was used to evaluate CUB-200-2011. The framework did not perform well with images than using

Dataset	FSL	Embed.	Meta Iter.	Accuracy(%)	Precision(%)	F1-score(%)
CUB-200-2011	5-way 10-shot	ResNet30	200	71.20 ± 0.13	74.70 ± 0.13	69.20 ± 0.13
Oxford-102	5-way 5-shot	ResNet30	200	84.80 ± 0.01	86.74 ± 0.02	84.59 ± 0.01
COVID19	3-way 5-shot	ResNet30	200	93.33 ± 0.01	94.44 ± 0.01	93.27 ± 0.01
Malaria	2-way 10-shot	ResNet30	300	83.33 ± 0.02	84.61 ± 0.03	83.21 ± 0.02

Results at a 95% of confidence interval (average accuracy, precision and F1-score ± standard deviation).

Table 6.2: Experiment results of GeMGF using only image data.

text data and the number of samples was increased from 5 to 10. The average F1-score of GeMGF with CUB-200-2911 was 69.20%. Oxford-102 was evaluated with 5-way 5-shot protocol, achieving 84.59% of average F1-score. The probable reason of better results with Oxford-102 is because with this dataset, five classes are randomly selected from 102 categories. However, in CUB-200-2011, five classes are randomly selected from double of classes, i.e., 200 classes, increasing the complexity for the algorithm predicting the class label.

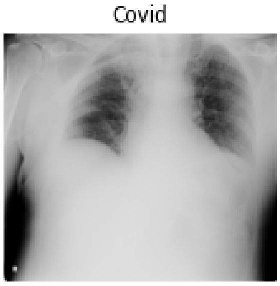
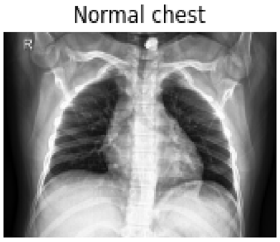
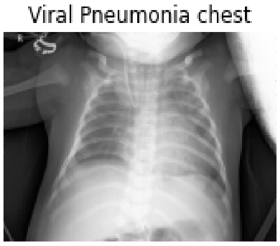
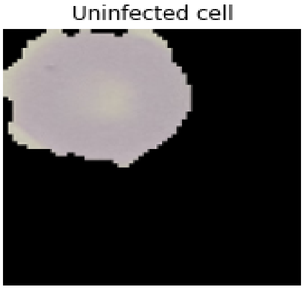
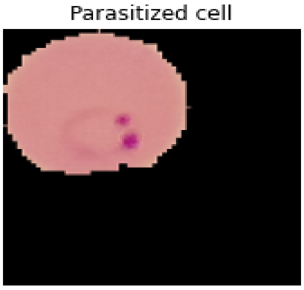
Dataset	Image examples		
COVID19	 	 	
Malaria	 	 	

Table 6.3: One example per class of COVID19 and Malaria datasets.

Next, we analyzed the two medical domain datasets. COVID19 is a small image dataset with 317 images of chest x-rays with three classes: normal chest x-ray, chest with COVID19, and chest with viral pneumonia, as illustrated in Table 6.3. This dataset was evaluated with 3-way 5-shot protocol, trained over 200 meta iterations, obtaining the average F1-score of 93.27%. Next, we analyzed the Malaria dataset containing 827 blood cell images categorized into two classes: parasitized cells and uninfected cells. Because of the low image quality of the blood cells, the Malaria dataset was trained longer (300 meta iterations) and used more samples for each class: 2-way 10-shot protocol. The result of our framework for the Malaria dataset was 83.21% of average F1-score. This result suggests that the performance of GeMGF for images relies on image quality.

6.2 Results of Multimodal Framework

This Section describes the experiments results of the multimodal version of GeMGF using image and text data. Our framework was evaluated using the multimodal benchmark datasets: CUB-200-2011 and Oxford-102. In this experiment, we analyzed the results of each dataset and also compared with the unimodal framework results.

The results are detailed in Table 6.4. Two FSL protocols were used: 5-way 5-shot and 5-way 1-shot, and the results are at a 95% of confidence interval. The first column represents the dataset, 'FSL' describes the data composition where 5-way 1-shot means five classes with one sample each. 'Meta Iter.' is the meta iteration, and the last three following columns are the metrics: average accuracy, precision and F1-score with the correspondent standard deviation. For both datasets, the image data was extracted with ResNet30 and the text data with BiLSTM.

Dataset	**FSL**	**Meta Iter.**	**Accuracy(%)**	**Precision(%)**	**F1-score(%)**
CUB-200-2011	5-way 5-shot	100	93.20 ± 0.07	93.80 ± 0.06	93.20 ± 0.07
CUB-200-2011	5-way 1-shot	100	85.60 ± 0.07	88.50 ± 0.06	85.50 ± 0.07
Oxford-102	5-way 5-shot	200	96.40 ± 0.01	97.00 ± 0.01	96.40 ± 0.01
Oxford-102	5-way 1-shot	200	94.80 ± 0.05	95.03 ± 0.06	94.42 ± 0.06

Results at a 95% of confidence interval (average accuracy, precision and F1-score ± standard deviation).

Table 6.4: Experiment results of multimodal GeMGF.

The multimodal GeMGF obtained 93.20% of average F1-score for the CUB-200-2011 dataset with 5-way-5-shot. The results for 5-way 1-shot was 85.50%, 9% lower than 5-way 5-shot. The outcome suggests that for a class-balanced dataset but with 200 different classes, our framework performs better with 5-shot or 5 samples than only one. The results of Oxford-102 was 96.40% of average F1-score with 5 shot and 94.42% with 1 shot, 2.09% lower. These results suggest that the multimodal framework performs better with Oxford-102 because five classes are selected from 102, lowering the difficulty of the algorithm predicting the class label. However, the Oxford-102 dataset had to be trained longer (200 meta iterations) because of the class-imbalanced composition.

The results of multimodal framework with CUB-200-2011 was 34.68% higher than the unimodal framework for image, and 0.1% higher than the unimodal framework for text. The results of multimodal framework with Oxford-102 was 13.96% higher than the unimodal framework for image, and 0.9% higher than the unimodal framework for text. These results suggest that the textual data helped the framework learn rich information to perform the multimodal classification.

6.3 Comparison with Baseline

This Section describes the comparison of the unimodal version of the framework. For this, two baseline models and other non-FSL models were used. We analyzed the results of each model, considering the dataset size, text length, sample distribution among classes, and the text language.

6.3.1 Comparison of Unimodal Framework for Text

Considering the lack of work with the FSL method using the seven datasets, we compare the results of GeMGF with models that use different architectures in Table 6.5. The first column refers to the model used in the comparison, followed by seven text datasets. These models do not use a FSL approach. Instead, they are trained using the entire dataset with the traditional mini-batch and epoch-based training procedures. For this, we used two baseline models according to the length of the text: CNN2L that comprises two layers of CNN used by [3] for short texts; and (ii) BiLSTM used by [4] for long texts. We used the version without a pre-trained Word2Vec for both baseline models. We also evaluated each dataset with the Transformer BERT [99] pre-trained over 110 million parameters.

Model	EN-T	Tweet250	JP-T	Livedoor	20NG	DEC6	CUB-200	Oxford-102
CNN2L[3]	75.90	47.20	70.21	-	-	-	-	-
BiLSTM[4]	-	-	-	85.76	84.40	92.00	88.33	67.53
BERT[99]	**78.00**	60.00	60.00	91.00	**89.00**	88.00	88.00	**88.00**
GeMGF	77.20	**89.20**	**94.98**	**93.75**	79.00	**95.00**	**92.00**	81.33

Table 6.5: Evaluation results (accuracy) of GeMGF and other models that adopt different approaches for text classification.

First, we analyzed the results of the three short text datasets (EN-T, Tweet250, and JP-T) evaluated using the CNN2L as a baseline model. Our framework obtained 77.20% of accuracy, 1.71% higher than CNN2L and 1.01% lower than BERT using the English dataset EN-T. The results of GeMGF was 89.20% of accuracy with Tweet250, 88.98% higher than CNN2L and 48.66% higher than BERT. For the Japanese dataset JP-T, the accuracy of our framework was 94.98%, 35.27% higher than the baseline and 58.30% higher than BERT. Comparing the results of GeMGF with the three short text datasets, it performed better with Japanese dataset, despite the smaller size of the dataset. This result is probably because Japanese sentence is formed by special characters called 'kanji' in which each character alone can denote one word. Figure 6.1 illustrates an example of a Japanese text from JP-T dataset. The highlighted text 出血熱 is a compound word with three 'kanjis'. Each 'kanji' has its own meaning: 出 means *come out or exit*; 血 means *blood*; and 熱 means *fever*. The three 'kanji' together mean *hemorrhagic fever*.

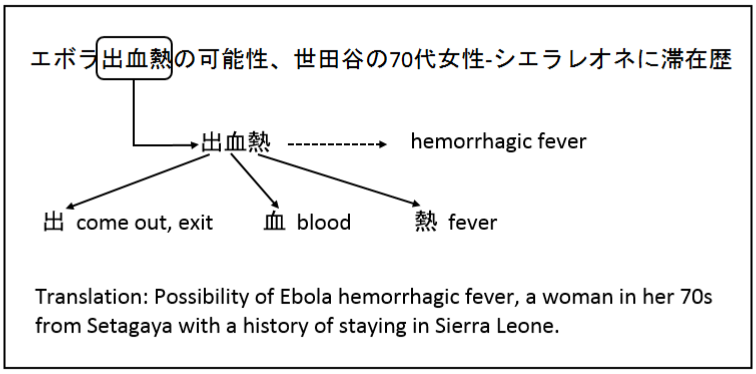

エボラ出血熱の可能性、世田谷の70代女性-シエラレオネに滞在歴
出血熱
hemorrhagic fever
出 come out, exit
血 blood
熱 fever
Translation: Possibility of Ebola hemorrhagic fever, a woman in her 70s from Setagaya with a history of staying in Sierra Leone.

Figure 6.1: Example of 'kanji'.

This result suggests that the embedding vector created from such rich feature may lead to a better vector representation then when created from alphabetic languages. This may be also the reason that the accuracy of JP-T is 9.76% higher than Tweet250, a English dataset, which has similar size and a class-balanced dataset.

Then we evaluated Livedoor, a large Japanese class-balanced dataset. Our framework achieved 93.75% accuracy classifying the text into four categories. This good result confirms that GeMGF may create a better vector representation for a non-alphabetic rich features.

Next, we analyzed the four long text datasets (20NG, DEC6, Oxford-102, and CUB-200) evaluated using the BiLSTM as a baseline model. In the 20NG dataset evaluation, four fixed classes out of 20 were used for simplicity and comparability. Our framework obtained 79.00% accuracy, 6.32% lower than the baseline. This result is probably because BiLSTM benefited from the knowledge learned from the average of 500 texts per class, while GeMGF learned incrementally from five texts from each class by episodes. The result of our framework was 12.65% lower than BERT using the 20NG dataset for the same reason. Moreover, the pre-trained knowledge of BERT on English public datasets may have boosted its good result. The accuracy of GeMGF using DEC6 (a Portuguese legal area dataset) was 3.26% higher than the baseline and 7.95% higher than BERT, probably because BERT was not pre-trained over Portuguese datasets.

CUB-200 was evaluated with BiLSTM because each text annotation is long text with up to 300 words. We also used the same five classes from a total of 200 to evaluate all models using CUB-200. The accuracy of GeMGF was 4.15% higher than BiLSTM and

4.45% higher than BERT. The baseline model and BERT had similar results with 88.33% and 88.00% accuracy, respectively. This result was contributed by the class-balanced composition of the dataset with an average of 60 texts per class. However, our framework outperformed with 92.00% of accuracy in the 5-way 5-shot FSL protocol.

Oxford-102 is composed of long texts with up to 120 words, and we evaluated it with BiLSTM baseline model. We used the same five classes out of 102 to assess all models using Oxford-102. The accuracy of GeMGF was 20.43% higher than BiLSTM probably because of the class-imbalanced composition of Oxford-102 with a minimum of 40 and maximum of 258 samples per class. The class with few samples contributed less to the result in BiLSTM, while in GeMGF, the FSL protocol of five samples per class incrementally trained by episodes had a positive effect in the accuracy. The result of our framework was 8.20% lower than BERT due to the previous knowledge that BERT learned from English pre-trained datasets.

Our unimodal framework outperformed the baseline model in three short text datasets and three long text datasets. Surprisingly, two Japanese datasets (JP-T and Livedoor) presented a high performance (94.98% and 93.75%) for the unimodal framework. Both results suggest that the rich 'kanji' feature representation contributes to the embedding vector's quality when using the FSL learning procedure. The framework had difficulties learning from short and noisy text (EN-T). Class-balanced long text (CUB-200) and short text (Tweet250) outperformed the baseline and BERT.

6.3.2 Comparison of Unimodal Framework for Image

Next, we evaluated the unimodal GeMGF for images with a baseline model and three other different architectures using four image datasets. The results are detailed in Table 6.5. The column 'Model' refers to the model architecture, 'Pixels' refers to the high and width of each image, and the following four columns represent the datasets. The metric used for the evaluation is accuracy. These models do not use a FSL approach. Instead, they are trained using the entire dataset with the traditional mini-batch and epoch-based training procedures. For this, we used the CNN3L as a baseline comprising three CNN layers. We also evaluated each dataset with three well-known computer vision models: (i) VGG-16 [51]; (ii) Inception V3 [21] ; and (iii) EfficientNet V2 [22]. VGG-16 [51] is a CNN model with 16 weight layers and 15 million parameters. Inception V3[21] also uses CNN, but with 19 weight layers and 22 million of parameters. EfficientNet V2 [22] uses a CNN architecture with 20 million parameters. The difference between EfficientNet V2 and the formers (VGG-16 and Inception V3) is that EfficientNet V2 is based on progressive learning. In this learning method, the model adaptively adjusts regularization according to the image size. All three computer vision models are pre-trained with ImageNet [100] - a large-scale public image dataset with 1.2 million images categorized into 1000 classes.

Model	Pixels	COVID19	Malaria	CUB-200-2011	Oxford-102
CNN3L	(180,180)	92.42	61.23	72.86	61.18
VGG-16 [51]	(224,224)	**93.94**	88.41	66.67	88.24
Inception V3 [21]	(299,299)	85.00	90.07	85.33	95.29
EfficientNet V2[22]	(384,384)	93.33	**91.91**	**93.33**	**95.29**
GeMGF	(180,180)	93.33	83.33	84.00	88.80

GeMGF used (180,180) pixels for all datasets, except for Malaria (224,224).

Table 6.6: Evaluation results (accuracy) of GeMGF and other models that adopt different approaches for image classification.

The first two datasets are from the medical domain (COVID19 and Malaria) and the other two are benchmark datasets (CUB-200-2011 AND Oxford-102). The best results are in bold. The results of GeMGF using COVID19 was 93.33% accuracy, 1.17% higher than the baseline model CNN3L, and our framework obtained similar results with VGG-16 (93.94%) and Efficient Net V2 (93.33%). Next, we analyzed the Malaria dataset. Our unimodal framework obtained 83.33% accuracy, 36.09% higher than the baseline model, and 10.23% lower than the best results of Efficient Net (91.91%). The low image quality of the Malaria dataset may have contributed to the results of GeMGF.

Next, we analyzed the results of the benchmark datasets. We used the same five classes out of 200 to evaluate all models using CUB-200-2011. Our unimodal framework obtained 84.00% accuracy with the CUB-200-2011 dataset, 15.28% higher than the baseline model, and 11.10% lower than the best result: Efficient Net V2 (93.33%). We also used the same five classes out of 102 to evaluate all models using Oxford-102. GeMGF result with Oxford-102 was 45.14% higher than the baseline but 7.30% lower than Efficient Net V2. The probable reason for Efficient Net V2 outperforming all models is that it takes advantage of the pre-trained knowledge obtained from ImageNet, which contains images of birds and flowers among the 1.2 million images.

The results suggest that the standard computer vision models have high performance influenced by the following factors: (i) the image data quality; (ii) the model uses a large pixel's size (Efficient Net V3); (iii) the model has sufficient data for training; and (iv) the model is pre-trained with a large-scale dataset (VGG-16, Inception V3, and EfficientNet V2). Our unimodal framework for image had similar results compared to the computer vision models only with the small size COVID19 dataset. However, GeMGF did not obtain the same good performance for the Malaria dataset, suggesting that our FSL-based framework is negatively influenced by the image data quality.

6.3.3 Comparison of Multimodal Framework

This Subsection describes the evaluation of the multimodal GeMGF compared with the state-of-the-art FSL models.

Authors	**Model**	**Accuracy (%)**
Vinyals et al., 2016 [66]	Matching Net	59.31
Finn et al., 2017 [58]	MAML	59.15
Snell et al., 2017 [6]	Prototypical Net	54.60
Zhao et al.,2021 [81]	CMKD	73.90 ± 0.01
Pahde et al.,2021 [2]	ProtoNet ResNet	85.30 ± 0.54
Li et al.,2021 [79]	ConvNet GCN	83.34 ± 0.56
Chen et al.,2021 [101]	Contextual Transfer	87.80
Xu et al.,2022 [102]	GCT	76.07
Ji et al.,2022 [78]	MAP-Net	88.30 ± 0.17
Munjal et al.,2023 [1]	QGN	91.86
Ours	**GeMGF**	**93.20 ± 0.07**

5-way 5-shot protocol (average accuracy(%) ± standard deviation).

Table 6.7: The comparison of GeMGF and other Multimodal FSL methods with CUB-200-2011

Table 6.7 details the models that use CUB-200-2011 dataset with 5-way 5-shot protocol. The first three models are the classical and well-known FSL models (Matching Net [66] and Prototypical Net [6]) and the meta-learning model (MAMAL [58]). The following seven works are recent models that use multimodal FSL combined with various methodologies. Our multimodal framework achieved the best result with 93.20% of average accuracy at a 95% of confidence interval.

Authors	**Model**	**Accuracy (%)**
Finn et al., 2017 [58]	MAML	79.00
Snell et al., 2017 [6]	Prototypical Net	89.20
Baneni et al.,2020 [103]	CNAPS	90.70 ± 0.50
Zhao et al.,2021 [81]	CMKD	50.30 ± 0.01
Pahde et al.,2021 [2]	ProtoNet ResNet	94.57 ± 0.13
Ji et al.,2022 [78]	MAP-Net	80.52 ± 0.16
Munjal et al.,2023 [1]	QGN	89.90
Ours	**GeMGF**	**96.40 ± 0.01**

5-way 5-shot protocol (average accuracy(%) ± standard deviation).

Table 6.8: The comparison of GeMGF and other Multimodal FSL methods for Oxford-102.

Table 6.8 details the models that use Oxford-102 dataset with 5-way 5-shot protocol. The first two models are the classical meta-learning model (MAMAL) and FSL model (Prototypical Net). Our multimodal framework achieved the best result with 96.40% of average accuracy at a 95% of confidence interval.

Non-FSL Methods

After comparing the multimodal GeMGF with other FSL models, we selected works that use different architectures and methods, such as models that rely on external knowledge and do not use FSL data sampling methods. Some recent works published on peer-reviewed platforms that use benchmark dataset CUB-200-2011 are listed in Table 6.9. The column 'Architecture' refers to the learning method or model used along with the pre-trained dataset.

Authors	Model	Architecture	Accuracy(%)
Yu et al., 2020 [76]	E-PGN [76]	ZSL	72.40
Bendre et al., 2021 [83]	M-VAE [83]	ZSL	62.90
Guang et al., 2022 [104]	CMSEA [104]	EfficientNetV2-S ImageNet21k	90.63
Chen et al., 2022 [105]	ACEN [105]	ResNet-101	89.70
Liu et al., 2022 [106]	TPSKG [106]	Vision Transformer ImageNet21k	**91.30**

Table 6.9: Recent works that adopt different approaches for multimodal classification using CUB-200-2011.

E-PGN [76] and M-VAE [83] use the multimodal ZSL method. However, instead of FSL, they use the entire dataset with mini-batches for the classification task. The main idea is to overcome the lack of data in one modality with complementary data from other modalities using GAN to create synthetic images. The other three models focus on complex computer vision models, such as CMSEA [104] which uses EfficientNet V2 pre-trained with ImageNet21k, ACEN [105] which uses ResNet-101 and TPSKG [106] based on Vision Transformer, pre-trained over large-scale dataset (ImageNet21k), with high computational resource consumption. TPSKG [106] obtained the best results with 91.30% accuracy.

Authors	Model	Architecture	Accuracy(%)
Yu et al., 2020 [76]	E-PGN [76]	ZSL	85.70
Chen et al., 2020 [107]		ZSL	46.70
Bendre et al., 2021 [83]	M-VAE [83]	ZSL	58.70
Liu et al., 2022 [106]	TPSKG [106]	Vision Transformer ImageNet21k	**99.50**
Fang et al., 2022 [80]	ACMR	ZSL	43.80

Table 6.10: Recent works that adopt different approaches for image classification using Oxford-102.

Table 6.10 details recent works evaluated with the Oxford-102 dataset, most of which use the ZSL method. The best performance was presented by TPSKG [106] with 99.50% accuracy, probably because it used the large-scale external knowledge of ImageNet21k.

6.4 Chapter Summary

This chapter describes the performance evaluation of the unimodal and multimodal GeMGF. The results of unimodal GeMGF were analyzed using six text datasets and four image datasets. The experiments of unimodal GeMGF using text data enabled us to analyze the framework dependency on the data quantity, quality, length of the text, data distribution among classes, and the language used in the text. For this, we used Japanese, English, and Portuguese multi-lingual datasets. Two Japanese datasets presented excellent performance suggesting that the rich non-alphabetic representation of 'kanji' contributes to the embedding vector quality using FSL procedure. Furthermore, the results suggest that the framework can adapt to text data belonging to various domains (legal area, online newsgroups about a variety of themes, and short text from Twitter about the epidemic). However, the framework had difficulties handling short and noisy texts.

The unimodal framework for text outperformed the baseline model in three short text datasets and four long text datasets. The class-balanced long text (CUB-200-2011) and short text (Tweet250) outperformed the baseline and BERT.

The experiments of unimodal GeMGF using image data were conducted using two benchmark datasets (CUB-200-2011 and Oxford-102) and two datasets from the medical domain (Malaria and COVID19). The computer vision model EfficientNet V2 outperformed all models using three datasets, which may be benefited from the pre-trained knowledge from ImageNet. The results of unimodal GeMGF for image had similar results compared to the computer vision models only with the small COVID19 dataset. These results suggest that the framework performance depends on the image data quality and quantity.

The multimodal framework was evaluated using two benchmark datasets (CUB-200-2011 and Oxford-102). The results suggest that text and image data combination helped the framework learn rich information and improve the overall performance.

Finally, the results of the multimodal GeMGF was compared with the state-of-the-art FSL models. Our framework outperformed Munjal et al. [1] by 1.43% with CUB-200-2011 and Pahde et al. [2] by 1.93% with Oxford-102.

Chapter 7

Analysis and Discussion

This chapter describes the impact assessment of some components in the multimodal GeMGF. Four ablation analyses were conducted by replacing or disabling the internal components. Section 7.1 describes the ablation analyses and the respective results compared to the multimodal GeMGF. Section 7.2 details the computer resource consumption. Section 7.3 discusses some relevant aspects of our research. Section 7.4 presents the summary of this chapter.

7.1 Ablation Analysis

This section describes the ablation analysis to evaluate the impact of four components in the multimodal GeMGF: (i) the Relation Network used in the multimodal FSL (sub-model 4); (ii) the impact of the image embedding (sub-model 1); (iii) the impact of the the text embedding (sub-model 2); and (iv) the impact of the multimodal fusion type used in the sub-model 1, sub-model 2, and sub-model 3.

7.1.1 Impact of the Relation Network (Ab1)

In this subsection, we analyzed the impact of the Relation Network in the multimodal FSL. For this, we evaluated our framework by replacing the Relation Network with the Euclidean distance metric, as illustrated in Figure 7.1. Using this metric to calculate the distance d between the class prototype and the query data, the probability distribution of one query data belonging to the class is calculated by Equation 2.13 detailed in Chapter 2. The negative distance $-d$ in Equation 2.13 indicates that the higher the value of $-d$, the lower the distance between the class prototype and the query data. For example, suppose that the distance d_1 of one query data from the prototype of class A is 100 and the distance d_2 of the same query data from the prototype of class B is 50. Comparing

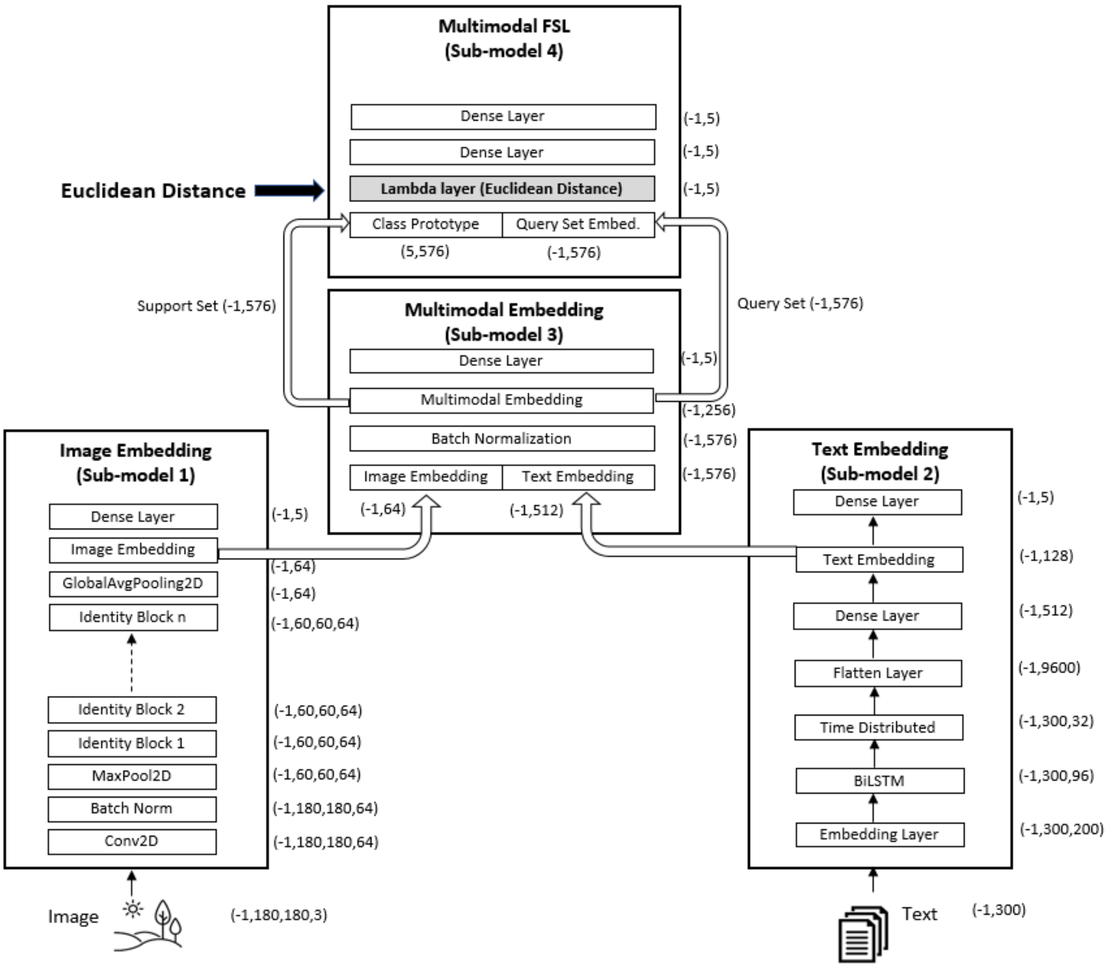


Figure 7.1: Multimodal Framework using Euclidean distance.

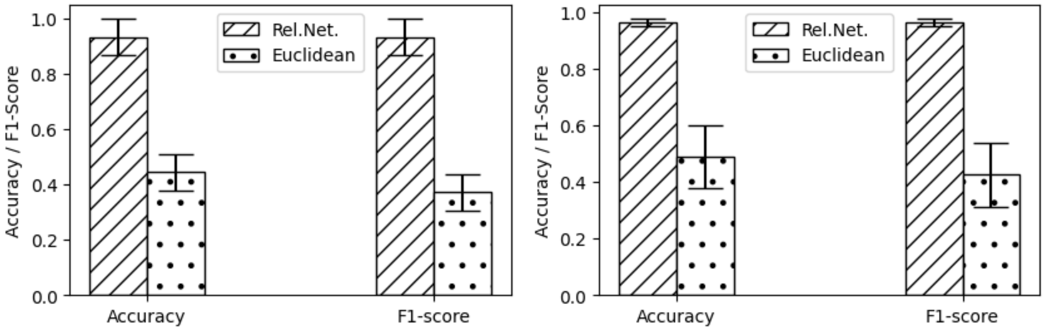


Figure 7.2: Impact of the Relation Network for multimodal GeMGF using CUB-200-2011 (left-hand side) and Oxford-102 (right-hand side).

the negative distance, $-d_2 > -d_1$, because -50 is greater than -100. Therefore, the distance d_2 is lower than d_1, and there is a higher probability of the query data belonging to class B than to class A.

Figure 7.2 illustrates the comparison results between the Relation Network and the

Euclidean distance. The bar with diagonal lines represents the Relation Network accuracy and F1-score, and the bar with black dots represents the results of the Euclidean distance. The left-hand side chart represents the result of CUB-200-2011 with 5-way 5-shot. The accuracy and F1-score using the Relation Network was 93.20%. The accuracy using the Euclidean distance was 44.40%, and the F1-score was 37.12%. The right-hand side chart illustrates the results for Oxford-102 with 5-way 5-shot. The accuracy and F1-score using the Relation Network was 96.40%. The accuracy of Euclidean distance was 48.80%, and the F1-score was 42.50%.

The results suggest that by using the Relation Network, our framework can efficiently learn the relation between the class prototype and the query set. The Euclidean distance represented a 52.36% accuracy decrease with CUB-200-2011 and a 49.37% accuracy decrease with Oxford-102 in our framework.

7.1.2 Impact of Image Data (Ab2)

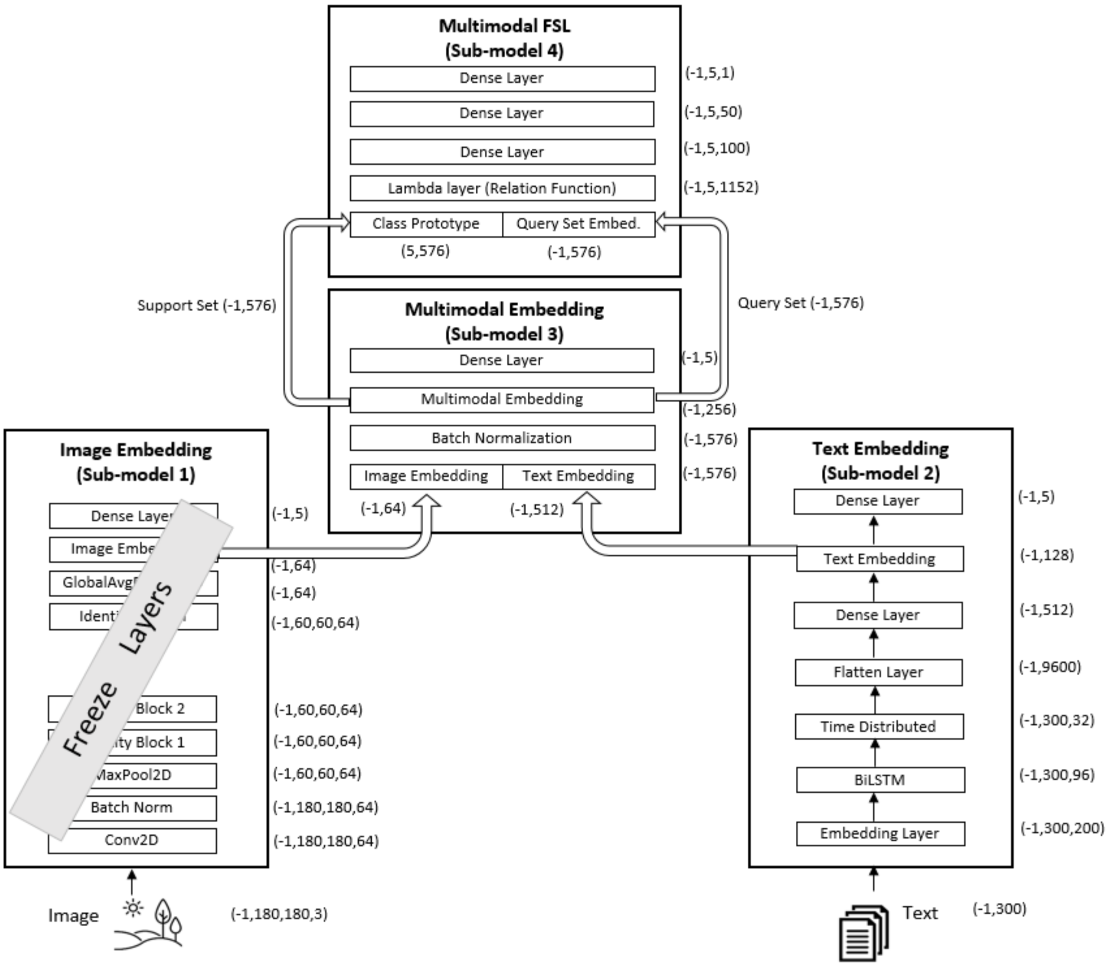


Figure 7.3: Multimodal Framework: freezing the image embedding layers.

The next ablation analysis evaluated the impact of the image data on the multimodal GeMGF. The evaluation was conducted by freezing the image embedding learnable layers, which means disabling the image embedding (sub-model 1) leaning functions. Concretely, all layers of the image embedding (sub-model 1) were frozen, as illustrated in Figure 7.3. The effect is that the image data are loaded and processed by sub-model 1, without weights optimization. The framework relies on the text embedding optimized by SGD, which is concatenated with the low-quality image embedding to learn the multimodal embedding. It is important to note that this ablation differs from the unimodal GeMGF for text, in which only text data was used. In the configuration of this ablation analysis, the framework receives both modalities as input data, but only the text data embedding is learned.

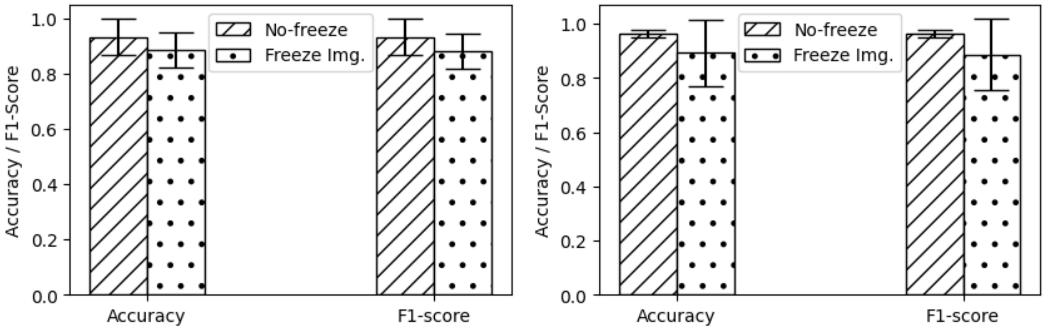


Figure 7.4: Impact of the image data by freezing image learning layers for multimodal FSL using CUB-200-2011 (left-hand side) and Oxford-102 (right-hand side).

The comparison results between the image/text no-freeze (multimodal GeMGF with all components) and the freeze image versions of GeMGF are illustrated in Figure 7.4. The bar with diagonal lines represents the accuracy and F1-score of the no-freeze version, and the bar with black dots represents the results of the freeze image version. The left-hand side chart illustrates the results of CUB-200-2011 with 5-way 5-shot. The accuracy and F1-score of the no-freeze version were 93.20%. The accuracy of the freeze image version was 88.40% and the F1-score was 88.10%. The right-hand side chart illustrates the results for Oxford-102 with 5-way 5-shot. The accuracy and F1-score of the no-freeze version were 96.40%. The accuracy of the freeze image was 89.20% and the F1-score was 88.66%.

The results suggest that by freezing the image layers, the impact using CUB-200-2011 is a 5.15% accuracy decrease and a 7.46% accuracy decrease using the Oxford-102 dataset. We did the following analysis for the low impact of freezing image layers. Both the text and image data contribute to the overall outcome of the multimodal GeMGF. The impact of freezing the image layers is not significant because the contribution of the image data was lower than the text data. The low contribution of the image data was caused, in turn, by the compact model designed for image embedding (sub-model 1). This sub-model is composed of only 30 identity blocks and 3 million parameters, and to avoid bias issues, we did not use a pre-trained model, such as ImageNet.

7.1.3 Impact of Text Data (Ab3)

Figure 7.5 illustrates the changes in our framework to analyze the impact of the text data on the multimodal GeMGF. The ablation analysis was executed by disabling the weight update of all layers of the text embedding (sub-model 2). This scenario differs from the unimodal GeMGF for image, in which only the image data was used. The effect of this ablation is that the text data are loaded, and processed by sub-model 2, without weight optimization. The framework relies on the image embedding optimized by SGD, which is concatenated with the low quality text embedding to learn the multimodal embedding.

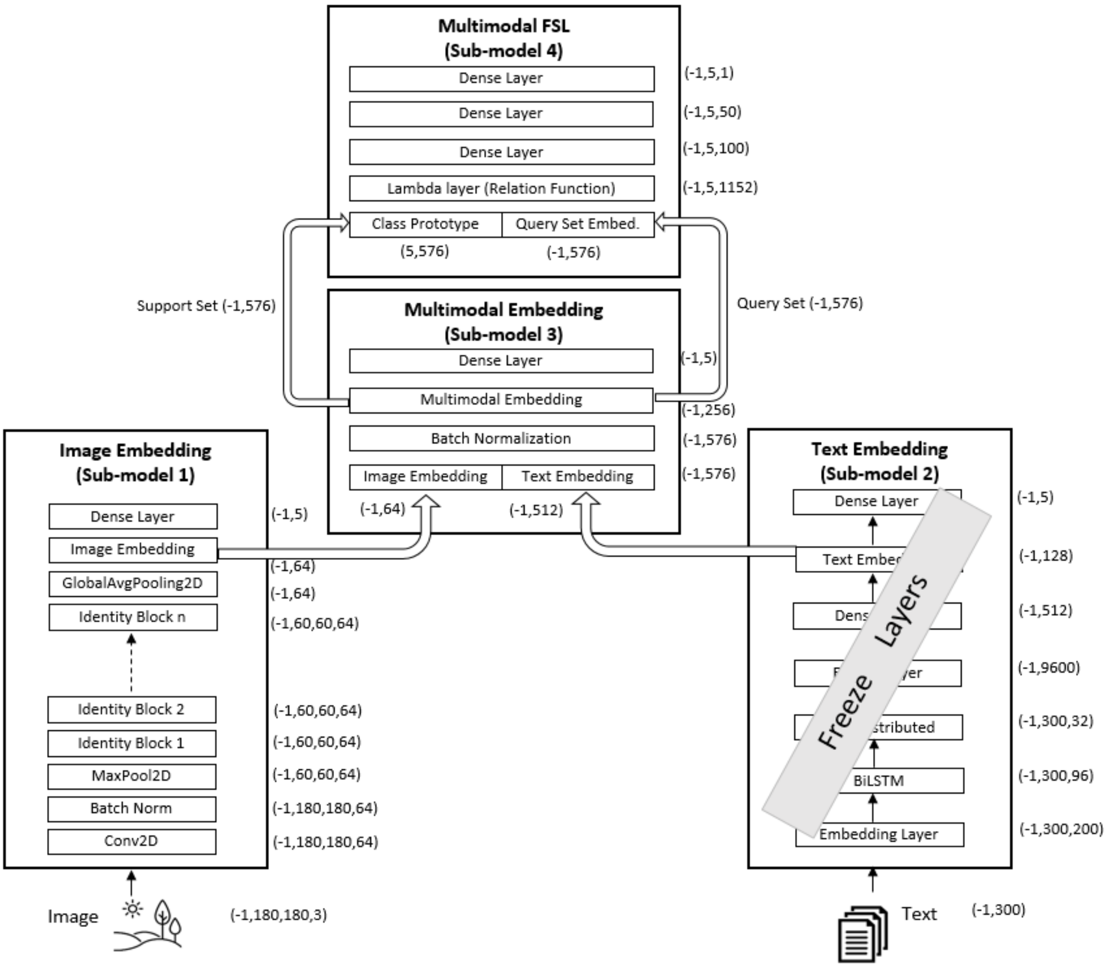


Figure 7.5: Multimodal Framework: freezing the text embedding layers.

The comparison between the image/text no-freeze and the text freeze versions of Generic Multimodal Gradient-Based Meta Framework (GeMGF) is illustrated in Figure

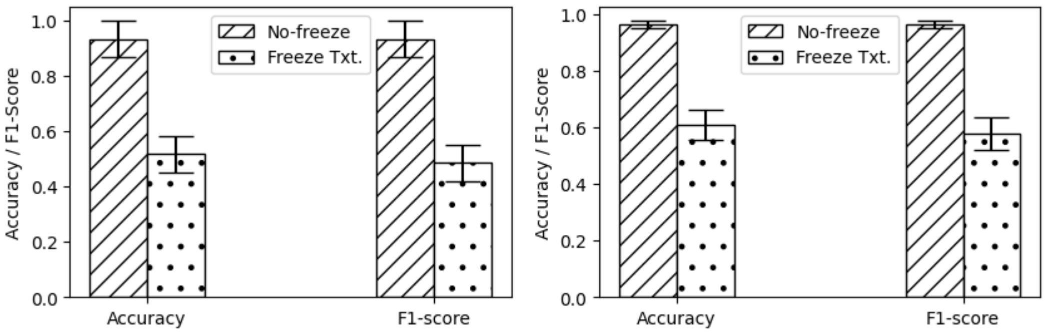


Figure 7.6: Impact of the text data by freezing the text learning layers for multimodal FSL using CUB-200-2011 (left side) and Oxford-102 (right side).

7.6. The left-hand side chart refers to the results of CUB-200-2011 with 5-way 5-shot. The accuracy and F1-score of the no-freeze version were 93.20%. The accuracy using the text freeze version was 51.16% and the F1-score was 48.50%. The right-hand side chart illustrates the results for Oxford-102 with 5-way 5-shot. The accuracy and F1-score using the no-freeze version were 96.40%. The accuracy and F1-score of the text freeze version were 60.80% .

The results suggest that the framework has a high impact by freezing the text embedding layers. The impact with CUB-200-2011 was a 45.10% accuracy decrease and a 36.92% accuracy decrease with Oxford-102. The experiment results depend on the quality and quality of text and image data. Therefore, using other multimodal datasets might have different outcomes.

7.1.4 Impact of Multimodal Fusion Type (Ab4)

The heterogeneous multimodal data need to be integrated to find the relationship between two or more modalities. Our framework uses the decision-level fusion. This means that the data from each modality are processed based on the modality-specific decision task and then integrated into the same feature space.

To evaluate the impact of the multimodal fusion type, the decision-level fusion was replaced by future-level fusion. For this, the classifiers of image and text embedding were disabled, as indicated by the dashed lines in Figure 7.7. In the feature-level fusion, the image and text data are integrated immediately after the extraction. Then the classifier of sub-model 3 is used in the multimodal embedding learning process.

The comparison between the decision-level fusion and the feature-level fusion on GeMGF is illustrated in Figure 7.8. The bar with diagonal lines represents the accuracy and F1-score of the decision-level fusion, and the bar with black dots represents the results of the feature-level fusion. The left-hand side chart refers to the results using

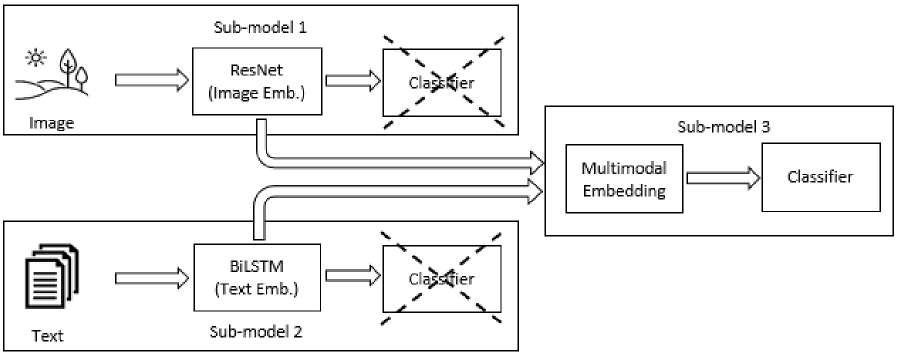


Figure 7.7: Feature-level multimodal fusion.

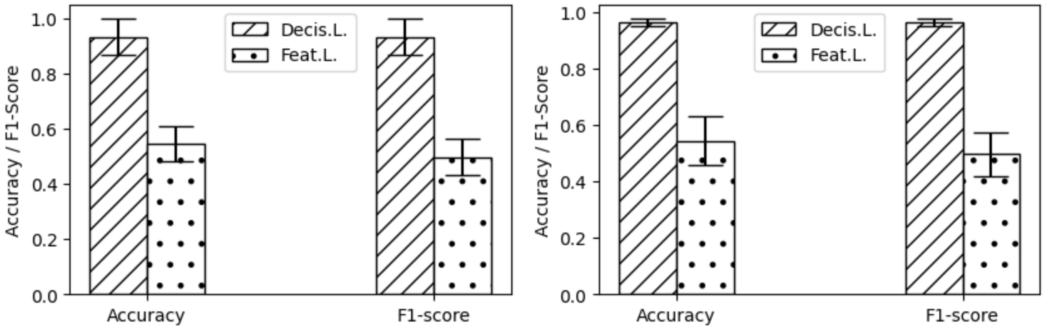


Figure 7.8: Impact of the multimodal fusion type using CUB-200-2011 (left-hand side) and Oxford-102 (right-hand side).

CUB-200-2011 with 5-way 5-shot. The accuracy and F1-score of the decision-level fusion were 93.20%. The accuracy using the feature-level fusion was 54.40% and the F1-score was 49.63%. The right-hand side chart illustrates the results using Oxford-102 with 5-way 5-shot. The accuracy and F1-score using the decision-level fusion was 96.40%. The accuracy using the feature-level fusion was 54.40% and F1-score was 49.61%.

The results suggest that by replacing the fusion type from the decision-level to the feature-level fusion, the framework has a high impact: 41.63% accuracy decrease on CUB-200-2011 and a 43.56% accuracy decrease on Oxford-102.

7.1.5 Analysis

In this section, four ablation analyses were described to verify the impact of disabling or replacing some components in the multimodal GeMGF.

The summarized results of the ablation analysis using CUB-200-2011 with 5-way 5-shot are detailed in Table 7.1. The ablation analysis that most impacted the framework was replacing the Relation Network with Euclidean distance (Ab1) with 44.40% accuracy. This outcome suggests that the Relation Network helps the model learn the relation between the class prototype and the query set efficiently. By freezing text embedding layers (Ab3), the accuracy was 51.16% while the full version of GeMGF was 93.20%, suggesting that the contribution of the text data is very high in the framework. On the other hand, freezing image layers (Ab2) had less impact on the results with 88.40% accuracy. The structure in which the image embedding (sub-model 1) was designed with only 3 million parameters has the benefit of a lightweight model. However, the drawback of a compact model is the low contribution to the multimodal embedding data. The image embedding (sub-model 1) can be improved by adding more identity blocks to the ResNet at the expense of increasing the computational cost. The results obtained replacing the decision-level fusion with future-level fusion (Ab4) was 54.40% accuracy, suggesting that the choice of the fusion type has a high impact on the multimodal framework.

Ablation	**Accuracy(%)**	**F1-score(%)**
Euclidean Distance (Ab1)	44.40 ± 0.09	37.12 ± 0.09
Freezing image layers (Ab2)	88.40 ± 0.09	88.10 ± 0.10
Freezing text layers (Ab3)	51.16 ± 0.09	48.50 ± 0.11
Future-level fusion (Ab4)	54.40 ± 0.04	49.63 ± 0.08
GeMGF	**93.20 ± 0.07**	**93.20 ± 0.07**

Table 7.1: Ablation analyses results of multimodal GeMGF using CUB-200-2011 with 5-way 5-shot.

The results of the four ablation analyses using CUB-200-2011 can be visualized in Figure 7.9. The box-plot helps us to analyze the distributional characteristics of a group of scores. The y axis represents the accuracy. In the x axis, 'GeMGF' refers to the box-plot of the multimodal framework with all components, 'Eucl.Dist' refers to the framework replacing Relation Network with Euclidean distance (Ab1), 'Freez.Img' refers to the modified framework by freezing image layers (Ab2), 'Freez.Text' refers to the modified framework by freezing text layers (Ab3), and 'Feat.Lev.' refers to the version of the framework replacing the decision-level with the feature-level fusion (Ab4).

It can be observed that the GeMGF and freezing image obtained high accuracy (0.96% on median for both). However, the interquartile range of GeMGF is shorter and less distributed than the box-plot of freezing image. This outcome suggests stability of the results when the framework uses all the components.

Next, we summarized the results of the ablation analyses using Oxford-102 with 5-way 5-shot in Table 7.2. The ablation analysis that most impacted the framework was using Euclidean distance (Ab1). Replacing the Relation Network with a linear classifier resulted in 48.80% accuracy, confirming the positive impact of the relation function on the framework. The results obtained replacing the decision-level fusion with future-level fusion (Ab4) was 54.40% accuracy, confirming the relevance of the fusion type on the multimodal

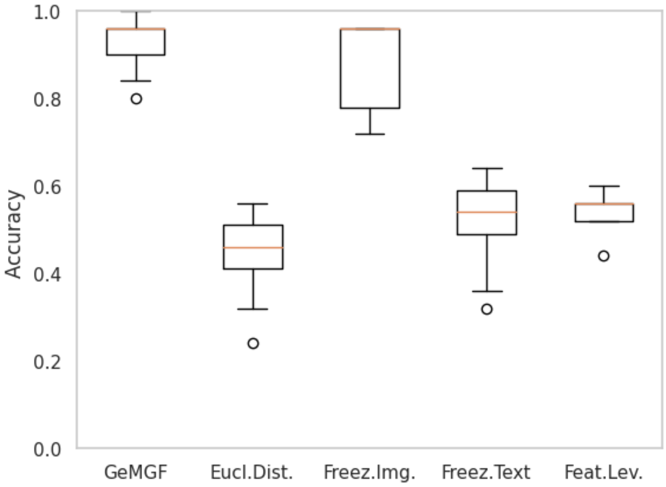


Figure 7.9: Ablation results of multimodal GeMGF using CUB-200-2011 with 5-way 5-shot.

framework. The impact of freezing text embedding layers was 60.80% accuracy. The impact of freezing image embedding layers was 89.20%, suggesting that the contribution of the text data is high in the framework.

Ablation	Accuracy(%)	F1-score(%)
Euclidean Distance (Ab1)	48.80 ± 0.11	42.50 ± 0.11
Freezing image layers (Ab2)	89.20 ± 0.12	88.66 ± 0.13
Freezing text layers (Ab3)	60.80 ± 0.05	60.80 ± 0.01
Future-level fusion (Ab4)	54.40 ± 0.08	49.61 ± 0.08
GeMGF	**96.40 ± 0.01**	**96.40 ± 0.01**

Table 7.2: Ablation results of multimodal GeMGF using Oxford-102 with 5-way 5-shot.

The distributional characteristics of the results using Oxford-102 can be visualized in Figure 7.10. GeMGF and freezing image version obtained high accuracy (0.96 and 0.92 on median, respectively). However, the interquartile range of GeMGF is shorter and less distributed than the box-plot of freezing image, which has more outliers. This outcome confirms the stability of the results when the framework uses all the components.

7.2 Computational Resource Consumption

The computational resources used by machine learning algorithms have grown as the models get more complex. In this scenario, the number of models trained on the cloud providers to access GPU and TPU has increased [108]. Recently, the environmental impact training large models have gained attention due to the growth of carbon emissions from data centers [109]. We found a few models listed in the selected works (Table

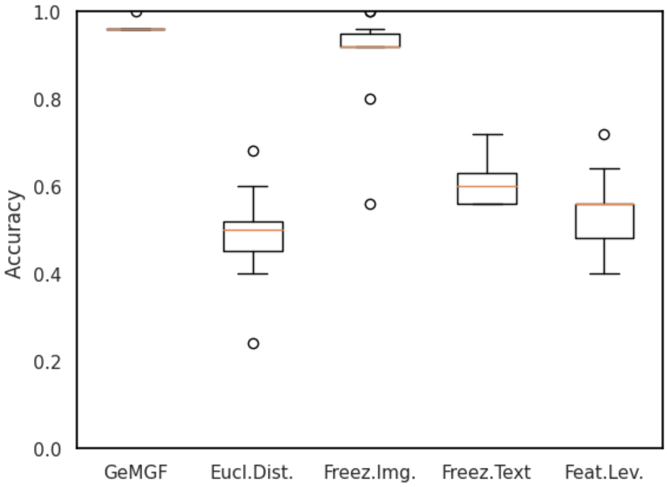


Figure 7.10: Ablation results of multimodal GeMGF using Oxford-102 with 5-way 5-shot.

3.4 of Chapter 3) that present information about the computational cost. The resource consumption can be measured at the software level (kernel sizes, number of parameters in a neural network, etc) and at the hardware level (processor, memory, IO peripherals, and others) [110]. For simplicity, we follow the analysis of Ji et al.[78] using the number of parameters and the floating point operations (FLOP). FLOP is a representative measure of CPU activity used as a basic computation unit.

The computational resource consumption of the unimodal and the multimodal GeMGF are described in Table 7.3. For this, we detailed the total number of parameters and the FLOP of our frameworks to compare with other models used for evaluation in Chapter 6. The FLOP of our framework was measured on Tesla T4 GPU using Keras-flops API [1].

Model	Parameter	FLOP
MAP-Net (2022) [78]	0.26M	7G
QGN (2023) [1]	11.54M	344,000G
Multimodal Transformer (2021) [30]	7,000M	-
Multimodal GeMGF	**14M**	**166G**
VGG-16 (2014) [51]	15M	-
Inception V3 (2016) [21]	22M	-
EfficientNet V2 (2021) [22]	24M	9G
Unimodal GeMGF for image	**4.5M**	**165G**
BERT (2018) [99]	110M	-
Unimodal GeMGF for text	**10M**	**0.10G**

M (million), G (Giga).

Table 7.3: Comparison of the computational resource consumption.

[1] https://pypi.org/project/keras-flops/

The multimodal GeMGF uses 14 million parameters, 99.8% less than the Multimodal Transformer [30], and 166 Giga FLOP for training, 99.9% less than QGN [1]. The unimodal GeMGF for image uses 4.5 million parameters, 81.25% less than the EfficientNet V2 [22]. The measured FLOP of the unimodal framework was 165 Giga, almost the same as the multimodal version. This number suggests that most of the computational unit operations are executed by the identity blocks of ResNet for image processing. This number also represents 94.5% more FLOP than EfficientNetV2 [22]. The unimodal GeMGF for text uses 10 million parameters, representing 90.90% fewer parameters than BERT[99]. The unimodal framework for text uses 0.10 Giga FLOP to process documents with BiLSTM, suggesting that in this framework, NLP tasks consume fewer computational units than computer vision tasks.

7.3 Discussion

This section discusses the results of the unimodal and multimodal framework and identifies some issues that need further analysis and investigation.

The multimodal GeMGF achieved excellent results using the two multimodal datasets when compared to the other state-of-the-art models, as described in Tables 6.7 and 6.8 of Chapter 6. However, the results of the multimodal GeMGF and the unimodal version for text are similar, as detailed in Table 7.4. The column 'Framework' identifies the multimodal and unimodal version, in which '(T)' stands for text and '(I)' for image.

Dataset	Framework	FSL	Accuracy(%)	Precision(%)	F1-score(%)
CUB-200-2011	Multimodal	5-way 5-shot	93.20 ± 0.07	93.80 ± 0.06	93.20 ± 0.07
	Unimodal (T)	5-way 5-shot	93.20 ± 0.03	94.10 ± 0.02	93.10 ± 0.03
	Unimodal (I)	5-way 10-shot	71.20 ± 0.13	74.70 ± 0.13	69.20 ± 0.13
Oxford-102	Multimodal	5-way 5-shot	96.40 ± 0.01	97.00 ± 0.01	96.40 ± 0.01
	Unimodal (T)	5-way 5-shot	95.60 ± 0.04	96.20 ± 0.03	95.55 ± 0.04
	Unimodal (I)	5-way 5-shot	84.80 ± 0.01	86.74 ± 0.02	84.59 ± 0.01

Results at a 95% of confidence interval (average accuracy, precision and F1-score ± standard deviation).

Table 7.4: Comparison between the multimodal and the unimodal framework.

The average accuracy of the multimodal and the unimodal version for text with CUB-200-2011 are the same (93.20%) and the average precision of the unimodal is slightly (0.03%) higher than the multimodal version. Based on these results, it seems that there is no benefit to using the multimodal version. We analyzed this outcome from two perspectives: (i) the CUB-200-2011 is a class-balanced benchmark dataset annotated with long texts that describes the image in detail. A real-world dataset may not have this high-quality text data; (ii) the image embedding (sub-model 1) could be improved, increasing the identity blocks and consequently increasing the multimodal framework accuracy. However, this change will increase the computation cost.

Our framework is designed to execute a task using a compact multimodal learning model. The supplementary and complementary information of different modalities help the overall framework performance without increasing the computational cost. For further analysis, we plan to evaluate our framework with a real-world multimodal dataset with possible class-imbalanced and low-quality text annotation or low-quality image dataset.

The results of the unimodal framework for image could be improved by pre-training the model with the widely used ImageNet. This external knowledge could boost the accuracy but can have a negative effect, such as model bias. Furthermore, if the image to be predicted is not in the ImageNet, the accuracy could have low or no improvement.

The unimodal framework for text using non-alphabetic languages achieved extraordinary results. At this moment of the research, we do not have the theoretical evidence for this outcome, and we plan further research.

Multimodal learning, especially using image and text, has made a great advance with Transformer-based models. The results of these models are not comparable with our work because they have different goals. Two distinct groups can be identified with the following analyses:

Group 1: Multimodal FSL models (including GeMGF):

- The goal is to learn from a few samples of multimodal data with no external knowledge;
- The model uses the episode-based k-class n-shot approach to mimic the real-world scenario;
- Prioritize compact and smalls model over the model performance.

Group 2: Models that handle complex computer vision tasks:

- The goal is to address complex computer vision problems using special architectures, such as Vision Transformer with fine-grained features and image transformation techniques;
- Use external knowledge to increase the model performance;
- Use the standard dataset division (training, testing, and evaluation) with mini-batches, increasing the hurdle for the model to predict one out of 150 or 200 classes in the case of CUB-200-2011.
- Prioritize the model's high performance over the model complexity and computational resource consumption.

Both groups still have open problems that need to be addressed, such as the high resource consumption and model bias issue.

7.4 Chapter Summary

This chapter describes the ablation analysis conducted to evaluate the impact of four components in the multimodal GeMGF: (i) the Relation Network (used in the sub-model 4); (ii) the image embedding (sub-model 1); (iii) the text embedding (sub-model 2); and (iv) the multimodal fusion type used in the sub-model 1, sub-model 2, and sub-model 3.

The component that has the most impact on the multimodal GeMGF is the Relation Network. The impact of the Relation Network was evaluated by replacing it with the Euclidean distance. The multimodal GeMGF obtained 109.90% higher accuracy compared to Euclidean distance with CUB-200-2011. The accuracy gain was 97.54% with Oxford-102. The results suggest that the Relation Network helps the framework learn more efficiently the relation between the class prototype and the query set than the Euclidean distance.

The impact of the image data on the multimodal GeMGF was evaluated by freezing the learnable layers of the image embedding (sub-model 1). In this ablation analysis, the accuracy decreased by 5.15% with CUB-200-2011 and 7.46% with Oxford-201.

The impact of the text data on the multimodal GeMGF was evaluated by freezing the learnable layers of the text embedding (sub-model 2). It was observed an accuracy decrease of 45.10% with CUB-200-2011 and 36.92% with Oxford-201.

The impact of the multimodal fusion method was conducted by replacing the decision-level fusion with feature-level fusion. The framework accuracy decreased by 41.63% with CUB-200-2011 and 43.56% with Oxford-102 using feature-level fusion.

The computational resource consumption to train our framework was measured using the number of parameters and the floating point operations (FLOP). We have found a few models listed in the selected works (Chapter 3) and in the evaluation (Chapter 6) that detail this information for comparison. The multimodal GeMGF uses 99.8% fewer parameters than the Multimodal Transformer [30]. The unimodal GeMGF for image uses 81.25% fewer parameters than the EfficientNet V2 [22], and the unimodal GeMGF for text 90.90% fewer parameters than BERT[99]. The FLOP to train the multimodal GeMGF was 99.9% less than QGN [1]. However, 94.5% more FLOP was measured for the unimodal GeMGF for image than EfficientNetV2 [22], indicating a need for further improvement.

The unimodal framework for text using Japanese achieved unexpectedly good results. This outcome suggests that using the character level separation between words enables a rich feature representation for Japanese texts. However, the behavior of our framework using non-alphabetic languages needs further analysis.

In the discussion section, some aspects of our research that still need further analysis were described. The first aspect is that we used two benchmark datasets to evaluate the multimodal GeMGF to enable comparison with other state-of-the-art works. The framework needs further analysis by using a real-world multimodal dataset with possible class-imbalanced and low-quality text annotation or low-quality image data. The second aspect is that the framework needs analysis from the model fairness perspective. The fact that it does not use a pre-trained model will not avoid language or image bias and fairness issues.

Chapter 8

Conclusion

The recent achievements of Transformer-based approaches have revolutionized machine learning in several areas. The architectures of these approaches with attention mechanism boosted research in NLP [24], computer vision [25], bio-medicine [28], and others. However, to achieve these excellent results, most models depend on large labeled datasets or rely on pre-trained models.

Due to the diversity of domain contexts in which these models are used, it is challenging to create models that learn from limited data, adapt, and generalize in the open-world scenario. For this, we devised the GeMGF framework. Using multimodal FSL, the model can be exposed to a more realistic scenario where limited labeled data are available for training.

The attention of the academic community to multimodal learning has grown fast in the last years. A broad systematic literature review was conducted, and 19 publications were selected from 138 works about multimodal FSL. The selected works use a diversity of methods, including GAN, GNN, ZSL, Transformer, and VAE. The detailed analyses of the selected works enabled the detection of the main advantages of each method, along with the remaining challenges and gaps in multimodal FSL. These factors helped us to design GeMGF.

We approached the problem of learning from a few data from two perspectives: model and data. Considering the model perspective, the model may have generalization problem in regular supervised learning if the unseen examples are not contained in the training dataset. This problem is addressed by meta-learning that uses two learning levels: a meta-learner and a base learner.

Considering the data perspective, the scarcity of data was compensated using multimodal learning, where complementary information of one modality can help the data representation. The main goal of multimodal learning is to create an abstraction of a unified representation of different modalities. Multimodal data representation is challenging because of the heterogeneity of data structures, sizes, and dimensions. In this process, the choice of multimodal data fusion type is relevant to find the relationship between two or more modalities.

The GeMGF framework is implemented with the base-leaner and the meta-learner. The base learner considers the data perspective and is responsible for extracting and representing the multimodal data. We used a modified ResNet30 for image feature extraction and BiLSTM for text feature extraction. We chose ResNet because the identity block composition is adaptable to the available computational resource, and we could keep the model compact with 30 identity blocks. The BiLSTM was used to capture the context of past and future time steps for long texts. After the feature extraction, the model learns the alignment between image and text data, integrates into the same feature space, and reduces the semantic gap between different modalities. We used the decision-level fusion, where each modality has an independent classifier resulting in a more flexible framework. Then the Prototypical Network combined with Relation Network is used to learn the relation between the class prototype and the query set.

In the meta-learner, the parameters of the base learner are updated periodically by using Reptile — an optimization-based meta-learning. The Reptile, jointly with FSL, helps to optimize the entire framework's learning capabilities. The overall framework configuration reduces the dependency on large annotated datasets.

In this thesis, in addition to the multimodal GeMGF, we implemented the unimodal version to evaluate the flexibility and adaptability of the framework in different scenarios. The evaluation of our framework was conducted using ten datasets from various domains and characteristics.

The unimodal framework for text was evaluated with eight text datasets. We used five real-world text datasets (EN-T, Tweet250, JP-T, Livedoor, and DEC6) to evaluate our framework with heterogeneous and challenging scenarios: (i) noisy short texts, (ii) legal domain long text, and (iii) multi-lingual texts. We also used three widely adopted benchmark text datasets (20NG, Oxford-102, and CUB-200-2011). Through the experiments, we analyzed the framework's dependency on the data quantity, quality, the data distribution between classes, and the languages used in the text.

The unimodal framework for text outperformed the baseline model in three short text datasets (EN-T, Tweet250, and JP-T) and four long text datasets (Livedoor, DEC6, CUB-200-2011, and Oxford-102). The class-balanced long text (CUB-200-2011) and short text (Tweet250) outperformed the baseline and Transformer BERT.

Two Japanese datasets presented excellent performance suggesting that the rich non-alphabetic representation of 'kanji' contributes to the embedding vector quality using FSL procedure. The results of unimodal framework for text suggest that the framework can adapt to different scenarios (legal area, online newsgroups about various themes, and short text from Twitter). However, the framework had difficulties handling short and noisy texts.

The unimodal framework for image was evaluated with two medical domain datasets (COVID19 and Malaria), and two benchmark datasets (Oxford-102 and CUB-200-2011). Our unimodal framework achieved similar results with EfficientNet V2 [22] only with COVID19 datasets. The computer vision model EfficientNet V2 has the advantage of pre-trained knowledge obtained from ImageNet, which contains images of birds and flowers among the 1.2 million images.

The multimodal framework was evaluated using two benchmark datasets (CUB-200-2011 and Oxford-102). The results suggest that text and image data combination helped the framework learn rich information and improve the overall performance. Our framework outperformed the state-of-the-art models: Munjal et al.[1] by 1.43% with CUB-200-2011 and Pahde et al. [2] by 1.93% with Oxford-102.

The ablation analyses were conducted to evaluate the impact of four components in the multimodal GeMGF: the Relation Network, the image embedding (sub-model 1), the text embedding (sub-model 2), and the multimodal fusion type.

The Relation Network was the component that most impacted the multimodal GeMGF. Our framework obtained 109.90% higher accuracy than the Euclidean distance with CUB-200-2011 and 97.54% with Oxford-102. This outcome suggests that Relation Network helps the model learn the relation between the class prototype and the query set efficiently.

The multimodal fusion method was the second component with a high impact on our framework. By replacing the decision-level fusion with feature-level fusion, the framework accuracy decreased by 41.63% with CUB-200-2011 and 43.56% with Oxford-102.

The third component that most impacted the framework is the text data, evaluated by freezing the learnable layers of the text embedding (sub-model 2). We observed an accuracy decrease of 45.10% with CUB-200-2011 and 36.92% with Oxford-201.

The component with the lowest impact on the multimodal GeMGF was the image data, evaluated by freezing the image embedding (sub-model 1) learnable layers. In this ablation analysis, the accuracy decreased by 5.15% with CUB-200-2011 and 7.46% with Oxford-201. This low impact on the framework is explained by the compact design of the image embedding (sub-model 1) with only 3 million parameters. This sub-model could be improved by adding more identity blocks to the ResNet and using external knowledge at the expense of increasing the computational cost.

The environmental impact training large models have gained attention due to the growth of carbon emissions from data centers [109]. Many academic and industry machine learning models are trained on cloud services. Our framework uses the cloud service of Google Colab. These models may collectively contribute to the carbon emissions increase, and we consider the effort to create compact models relevant. We measured the computer resource consumption of the framework using two features: the number of parameters and the floating point operations (FLOP). The multimodal GeMGF uses 14 million parameters, 99.8% less than the Multimodal Transformer [30].

In this last chapter, we summarized the methods used in GeMGF, comparing the results obtained from the empirical evaluation with the state-of-the-art models. To conclude this thesis, Section 8.1 describes how we accomplished the research objectives, and Section 8.2 details the contributions of our work. Section 8.3 describes some aspects of this thesis that need further research.

8.1 Addressing the research objectives

This section describes how each research objective were addressed.

Reduce the cost to annotate large datasets - Generally, the performance of the supervised machine learning models depend on large labeled dataset. GeMGF uses FSL method to learn from a few data and reduce the cost of annotating large datasets. The framework was evaluated using ten datasets: six text datasets, two image datasets, and two multimodal (image/text) datasets (Table 5.1 of Chapter 5). Among the text datasets, four (EN-T, Tweet50, JP-T, and DEC6) are small in size with less than 1200 data. The two image datasets (COVID19 and Malaria) have less than 830 data. The multimodal datasets (CUB-200-2011 and Oxford-102) are medium size with 8,189 and 11,788 data, respectively. However, the framework used FSL protocol with 5-way or five classes for training reducing the amount of data requirement. In other words, the framework used an average of 300 multimodal data with CUB-200-2011 and 400 with Oxford-102. Analyzing the size of the datasets used in this work, except for Livedoor and 20NG, with the size of 4,572 and 11,314, respectively, the size of the eight datasets used by the GeMGF for training can be considered small. Considering the four small size text datasets (EN-T, Tweet50, JP-T, and DEC6), the unimodal framework outperformed the baseline and BERT with three text datasets (Tweet50, JP-T, and DEC6) (Table 6.5 of Chapter 6). Considering the image datasets (COVID19 and Malaria), our unimodal framework outperformed the baseline model, as described in Table 6.6. The multimodal framework outperformed the state-of-the-art model of Munjal et al. [1] by 1.43% with CUB-200-2011 and Pahde et al. [2] by 1.93% with Oxford-102, as described in Table 6.7 and Table 6.8 of Chapter 6. The evaluation results suggest that the unimodal and multimodal frameworks can reduce the dependency on large labeled datasets.

Reduce the semantic gap between different modalities - Our framework uses a modified ResNet30 for image extraction and BiLSTM for text extraction. After the feature extraction, the multimodal embedding (sub-model 3) learns the alignment between image and text data, then integrates it into the same feature space. We used the decision-level fusion, where each modality has an independent decision task. With this architecture, the multimodal GeMGF achieved 93.20% accuracy for CUB-200-2011 and 96.40% accuracy for the Oxford-102 dataset. In this process, the image and text data helped to create a good vector representation for the multimodal GeMGF. The impact of freezing the image embedding layers (sub-model 1) resulted in a 5.15% accuracy decrease with CUB-200-2011 and a 7.46% accuracy decrease with Oxford-102 (Subsection 7.1.2 of Chapter 7). The impact of freezing the text embedding layers was higher, 45.10% accuracy decrease with CUB-200-2011 and a 36.92% accuracy decrease with Oxford-102 (Subsection 7.1.3 of Chapter 7). The results suggest that the multimodal embedding (sub-model 3) integrated the heterogeneous image and text data and efficiently learned the multimodal embedding vector.

By replacing the decision-level fusion with feature-level fusion, the accuracy drops by 41.63% with CUB-200-2011 and 43.56% with Oxford-102 (Subsection 7.1.4 of Chapter 7). This result suggests that after learning a good embedding, the semantic gap between image and text data can be minimized using the decision-level fusion.

Create a model that learns from a few data preserving previously learned knowledge - GeMGF uses FSL, in which the dataset is split into a support set and a query set. The training procedure occurs incrementally, in an episodic way. In each training episode, K labeled samples of the classes in the support set are randomly selected for training. In the test phase, few samples of the classes are selected from the query set. The class prototype is calculated based on the support set. Then the framework uses the Relation Network to learn the relation score between the class prototype and the data in the query set. The framework uses the relation score to predict whether the multimodal query data and the class prototype are from matching categories or not. With this architecture, the multimodal GeMGF achieved 93.20% accuracy for CUB-200-2011 and 96.40% accuracy for the Oxford-102 dataset. The impact of the Relation Network was evaluated by replacing it with a fixed metric: the Euclidean distance. We observed a 52.36% of accuracy reduction with CUB-200-2011 and a 49.37% of accuracy reduction with

Oxford-102 by replacing the Relation Network with the Euclidean distance (Subsection 7.1.1 of Chapter 7).

This result suggests that using the relation function instead of a linear classifier (such as Euclidean distance), the model can benefit from a learnable non-linear approach rather than a fixed metric. The Prototypical Network and the Relation Network combination helped the framework overcome the lack of training data.

CUB-200-2011 is a class-balanced dataset with an average of 60 data per class, and Oxford-102 has an average of 80 data per class. The unimodal GeMGF was evaluated with smaller text datasets outperforming BERT. DEC6 has an average of 20 samples per class, and the accuracy of the unimodal GeMGF was 7.98% higher than BERT. JP-T has 39 samples per class and outperformed BERT in 58.30% (Table 6.5 of Chapter 6).

The results of the multimodal and unimodal framework suggest that FSL can help the model efficiently learn from a few data.

Create a compact model to reduce the growth of computational cost - The computational cost to train large-scale models has grown dramatically in the past few years. Most of these complex models are trained on the cloud providers' data centers, which collectively may increase carbon emission. In the systematic literature review, we have found a few publications of multimodal FSL models that describe the information of the computational cost. For comparison purposes, we followed the strategy of Ji et al. [78] using the number of the model parameters and the FLOP.

The multimodal GeMGF uses 14 million parameters, 99.8% less than the Multimodal Transformer [30], and 166 Giga FLOP for training, 99.9% less than QGN [1]. The unimodal GeMGF for image uses 4.5 million parameters, 81.25% less than the EfficientNet V2 [22]. The measured FLOP of the unimodal framework was 165 Giga, 94.5% more than EfficientNetV2 [22]. The unimodal GeMGF for text uses 10 million parameters, representing 90.90% fewer parameters than BERT[99].

The results of multimodal and unimodal frameworks suggest that considering the number of parameters, our frameworks are smaller and more compact when compared to other models. However, there are still opportunities for improvement regarding FLOP efficiency.

8.2 Main contributions

This section describes the main contributions of our work:

1. A novel multimodal framework with Few-Shot learning that can alleviate performance degradation trained over a limited and a few samples of data.

To the best of our knowledge, the multimodal GeMGF is a novel approach that uses the combination of the following methods: a decision-level fusion for multimodal data integration, the Relation Network combined with Prototypical Network for FSL, and an optimization-based meta-learning method.

2. GeMGF is trained end-to-end from scratch, avoiding possible language bias and fairness issues of pre-trained models.

   We could have trained the image embedding (sub-model 1) of the multimodal GeMGF with ImageNet to improve the overall framework outcome because ImageNet contains examples of flowers and birds. The negative effect is the possible gender, race, age, and other fairness issues influenced by ImageNet. Similarly, the text embedding (sub-model 2) could have been pre-trained with Word2Vec or a Transformer model, resulting in possible language bias from the Word2Vec or from the large-scale public dataset on which the Transformer was pre-trained. All the four sub-models that compose the multimodal framework and the two sub-models of unimodal framework are trained from scratch. Despite not using external knowledge from the pre-trained models, our unimodal and multimodal framework achieved excellent results.

3. The framework has independent multimodal feature extractors adaptable to other architectures.

   The multimodal framework uses ResNet for image extraction in the sub-model 1 and BiLSTM for text extraction in the sub-model 2. Both sub-models are independent and can be easily replaced by other model architectures. This independence is demonstrated in the unimodal framework. The text extraction was replaced by CNN to process short text datasets (EN-T, Tweet250, and JP-T), as described in Subsection 5.2.2 of Chapter 5 and illustrated in Figure 5.2. Then the CNN was replaced by BiLSTM to process long text datasets (Livedoor, 20NG, DEC6, CUB-200, and Oxford-102). Our framework outperformed BERT in five (Tweet250, JP-T, Livedoor, DEC6, and CUB-200) out of eight text datasets, as detailed in Section 6.3.1 of Chapter 6. The unimodal framework can easily be adapted for image extraction, as illustrated in Figure 5.3.

4. The framework has possibilities for applications in various domains.

   We chose data from various domains to evaluate the framework's adaptability and flexibility. The unimodal framework was evaluated with text data from the legal area, online newsgroups about various themes, and short texts from Twitter about the epidemic. The image data are related to the medical area: chest x-ray images and blood cell images. The multimodal data are benchmark datasets from the botanical and zoological areas. Our framework can be used in different domains because it does not rely on external knowledge.

5. The unimodal framework for text is multilingual and adaptable to alphabetic and non-alphabetic languages.

   The adaptability of our framework for text was evaluated using three different languages: Portuguese, English, and Japanese. We used eight text datasets, of which five are English texts, two are Japanese texts, and one is Portuguese text. The text data from alphabetic languages (Portuguese and English) were extracted using a word-level approach with space as a word separator. Because non-alphabetic languages, such as Japanese and Chinese, do not have space separation between words, we used the character-level approach for the text extraction. The unimodal framework for text outperformed the baseline model and BERT in two English datasets (Tweet250 and CUB-200-2011), two Japanese datasets (JP-T and Livedoor), and one Portuguese dataset (DEC6).

8.3 Feature Works

- **Framework enhancement** - Despite the promising results of GeMGF, the framework focus on multi-class categorization task. We plan to enhance its applicability to image captioning and multi-label classification.
- **Explainable model** - Pre-trained models that use large-scale image and text information are more prone to present fairness and bias issues. GeMGF does not use external knowledge in the training procedure. However, this aspect is insufficient for the model to avoid language bias and guarantee fairness in the outcomes. To address this problem, we need to analyze why the model achieved a particular prediction, which input features most contributed to that decision, and identify a possible model bias [71]. An explainable model will provide trust and transparency.
- **Sustainable model** - Complex models with billions of parameters are high-computer resource consumers. The GPU, TPU, and memory used during the long training time leads to more CO2 emission than compact and small models. Our framework was designed to be compact, with 14 million parameters. However, we observed that there are still opportunities for improvement in the FLOP efficiency.

- **Multi-lingual model** - The unimodal framework for text presented excellent results with the Japanese dataset. We plan further research to analyze the multimodal FSL with non-alphabetic languages.
- **Open-world scenario** - Our future work will cover the open-world scenario, considering multimodal new class identification and outliers detection.

Finally, in the academic research, we usually train our model over a static dataset. However, in the real-world scenario, data are dynamic, change over time, and more research should address adaptable and flexible models to fill this gap.

References

[1] Munjal, Bharti, Alessandro Flaborea, Sikandar Amin, Federico Tombari, and Fabio Galasso: *Query-guided networks for few-shot fine-grained classification and person search.* Pattern Recognition, 133:109049, 2023. ix, 21, 22, 23, 27, 28, 65, 67, 77, 78, 80, 84, 85, 87

[2] Pahde, Frederik, Mihai Puscas, Tassilo Klein, and Moin Nabi: *Multimodal prototypical networks for few-shot learning.* In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2644–2653, 2021. ix, 21, 22, 23, 26, 28, 44, 65, 67, 84, 85

[3] Enamoto, Liriam, Li Weigang, and Geraldo P Rocha Filho: *Generic framework for multilingual short text categorization using convolutional neural network.* Multimedia Tools and Applications, 80(9):13475–13490, 2021. xiv, 7, 33, 41, 61

[4] Enamoto, Liriam, Andre RAS Santos, Ricardo Maia, Li Weigang, and Geraldo P Rocha Filho: *Multi-label legal text classification with bilstm and attention.* International Journal of Computer Applications in Technology, 68(4):369–378, 2022. xiv, 1, 9, 41, 43, 61

[5] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun: *Deep residual learning for image recognition.* In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. xiv, 9, 10, 31, 32

[6] Snell, Jake, Kevin Swersky, and Richard Zemel: *Prototypical networks for few-shot learning.* In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4080–4090, Red Hook, NY, USA, 2017. Curran Associates Inc., ISBN 9781510860964. xiv, 11, 12, 36, 65

[7] Sung, Flood, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales: *Learning to compare: Relation network for few-shot learning.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. xiv, 11, 12, 13, 14, 36, 37

[8] Huang, Gang: *E-commerce intelligent recommendation system based on deep learning.* In *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pages 1154–1157. IEEE, 2022. 1

[9] Choudhury, Sasmita Subhadarsinee, Sachi Nandan Mohanty, and Alok Kumar Jagadev: *Multimodal trust based recommender system with machine learning ap-*

proaches for movie recommendation. International Journal of Information Technology, 13(2):475–482, 2021. 1

[10] Van Belle, Rafaël, Bart Baesens, and Jochen De Weerdt: *Catchm: A novel network-based credit card fraud detection method using node representation learning.* Decision Support Systems, 164:113866, 2023, ISSN 0167-9236. `https://www.sciencedirect.com/science/article/pii/S0167923622001373`. 1

[11] Khan, Ameer Tamoor, Xinwei Cao, Shuai Li, Vasilios N Katsikis, Ivona Brajevic, and Predrag S Stanimirovic: *Fraud detection in publicly traded us firms using beetle antennae search: A machine learning approach.* Expert Systems with Applications, 191:116148, 2022. 1

[12] Lipyanina, Hrystyna, Valeriya Maksymovych, Anatoliy Sachenko, Taras Lendyuk, Andrii Fomenko, and Ivan Kit: *Assessing the investment risk of virtual it company based on machine learning.* In *International Conference on Data Stream Mining and Processing*, pages 167–187. Springer, 2020. 1

[13] Dai, Zhiyong, Jianjun Yi, Lei Yan, Qingwen Xu, Liang Hu, Qi Zhang, Jiahui Li, and Guoqiang Wang: *Pfemed: Few-shot medical image classification using prior guided feature enhancement.* Pattern Recognition, 134:109108, 2023. 1

[14] Dara, Suresh, Swetha Dhamercherla, Surender Singh Jadav, CH Babu, and Mohamed Jawed Ahsan: *Machine learning in drug discovery: a review.* Artificial Intelligence Review, 55(3):1947–1999, 2022. 1

[15] Tang, Xiaolin, Kai Yang, Hong Wang, Jiahang Wu, Yechen Qin, Wenhao Yu, and Dongpu Cao: *Prediction-uncertainty-aware decision-making for autonomous vehicles.* IEEE Transactions on Intelligent Vehicles, 7(4):849–862, 2022. 1

[16] Cortes, Corinna and Vladimir Vapnik: *Support-vector networks.* Machine learning, 20(3):273–297, 1995. 1

[17] Genkin, Alexander, David D Lewis, and David Madigan: *Large-scale bayesian logistic regression for text categorization.* technometrics, 49(3):291–304, 2007. 1

[18] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner: *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 86(11):2278–2324, 1998. 1, 16

[19] Mikolov, Tomáš, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur: *Recurrent neural network based language model.* In *Eleventh annual conference of the international speech communication association*, 2010. 1

[20] Krizhevsky, Alex: *One weird trick for parallelizing convolutional neural networks.* arXiv preprint arXiv:1404.5997, 2014. 1

[21] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna: *Rethinking the inception architecture for computer vision.* In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1, 9, 63, 64, 77

[22] Tan, Mingxing and Quoc Le: *Efficientnetv2: Smaller models and faster training.* In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 1, 9, 63, 64, 77, 78, 80, 84, 87

[23] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin: *Attention is all you need.* In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 17

[24] Lu, Yu, Jiajun Zhang, Jiali Zeng, ShuangZhi Wu, and Chengqing Zong: *Attention analysis and calibration for transformer in natural language generation.* IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022. 2, 82

[25] Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, *et al.*: *An image is worth 16x16 words: Transformers for image recognition at scale.* arXiv preprint arXiv:2010.11929, 2020. 2, 82

[26] Kim, Wonjae, Bokyung Son, and Ildoo Kim: *Vilt: Vision-and-language transformer without convolution or region supervision.* In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 2

[27] Hao, Weituo, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao: *Towards learning a generic agent for vision-and-language navigation via pre-training.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146, 2020. 2

[28] Shah, Syed Muazzam Ali, Semmy Wellem Taju, Quang Thai Ho, Yu Yen Ou, *et al.*: *Gt-finder: Classify the family of glucose transporters with pre-trained bert language models.* Computers in Biology and Medicine, 131:104259, 2021. 2, 82

[29] Zhang, Xu Yao, Cheng Lin Liu, and Ching Y Suen: *Towards robust pattern recognition: a review.* Proceedings of the IEEE, 108(6):894–922, 2020. 2

[30] Tsimpoukelli, Maria, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill: *Multimodal few-shot learning with frozen language models.* Advances in Neural Information Processing Systems, 34:200–212, 2021. 2, 21, 22, 23, 24, 25, 77, 78, 80, 85, 87

[31] Zhu, Xizhou, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai: *Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16804–16815, 2022. 2, 21, 22, 23, 24, 25

[32] Song, Ge and Xiaoyang Tan: *Real-world cross-modal retrieval via sequential learning.* IEEE Transactions on Multimedia, 2020. 2, 3, 21, 22, 23, 25, 28, 29, 33

[33] Fan, Jiawei, Zhonghong Ou, Xie Yu, Junwei Yang, Shigeng Wang, Xiaoyang Kang, Hongxing Zhang, and Meina Song: *Episodic projection network for out-of-distribution detection in few-shot learning.* In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3076–3082. IEEE, 2022. 2, 21, 22, 23, 27, 28

[34] Kaur, Parminder, Husanbir Singh Pannu, and Avleen Kaur Malhi: *Comparative analysis on cross-modal information retrieval: a review.* Computer Science Review, 39:100336, 2021. 2, 31

[35] Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, *et al.*: *Overcoming catastrophic forgetting in neural networks.* Proceedings of the national academy of sciences, 114(13):3521–3526, 2017. 3

[36] Hospedales, Timothy, Antreas Antoniou, Paul Micaelli, and Amos Storkey: *Meta-learning in neural networks: A survey.* arXiv preprint arXiv:2004.05439, 2020. 3

[37] LeCun, Yann, Yoshua Bengio, *et al.*: *Convolutional networks for images, speech, and time series.* The handbook of brain theory and neural networks, 3361(10):1995, 1995. 6

[38] Sayadi, Hossein, Yifeng Gao, Hosein Mohammadi Makrani, Jessica Lin, Paulo Cesar Costa, Setareh Rafatirad, and Houman Homayoun: *Towards accurate run-time hardware-assisted stealthy malware detection: a lightweight, yet effective time series cnn-based approach.* Cryptography, 5(4):28, 2021. 6

[39] Fernandes Jr, Francisco E, Luis Gustavo Nonato, and Jó Ueyama: *A river flooding detection system based on deep learning and computer vision.* Multimedia Tools and Applications, 81(28):40231–40251, 2022. 6

[40] Johnson, Rie and Tong Zhang: *Effective use of word order for text categorization with convolutional neural networks.* arXiv preprint arXiv:1412.1058, 2014. 7

[41] Wang, Jin, Zhongyuan Wang, Dawei Zhang, and Jun Yan: *Combining knowledge with deep convolutional neural networks for short text classification.* In *IJCAI*, pages 2915–2921, 2017. 7

[42] Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin: *A neural probabilistic language model.* Journal of machine learning research, 3(Feb):1137–1155, 2003. 7

[43] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean: *Distributed representations of words and phrases and their compositionality.* In *Advances in neural information processing systems*, pages 3111–3119, 2013. 7, 26

[44] Caragea, Cornelia, Adrian Silvescu, and Andrea H Tapia: *Identifying informative messages in disaster events using convolutional neural networks.* In *International Conference on Information Systems for Crisis Response and Management*, pages 137–147, 2016. 7

[45] Elman, Jeffrey L: *Finding structure in time.* Cognitive science, 14(2):179–211, 1990. 8

[46] Hochreiter, Sepp: *The vanishing gradient problem during learning recurrent neural nets and problem solutions.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02):107–116, 1998. 8, 10, 25, 32, 33

[47] Hochreiter, Sepp and Jürgen Schmidhuber: *Long short-term memory.* Neural computation, 9(8):1735–1780, 1997. 8, 16

[48] Schuster, Mike and Kuldip K Paliwal: *Bidirectional recurrent neural networks.* IEEE transactions on Signal Processing, 45(11):2673–2681, 1997. 8, 16, 33

[49] Qin, Ya, Guo wei Shen, Wen bo Zhao, Yan ping Chen, Miao Yu, and Xin Jin: *A network security entity recognition method based on feature template and cnn-bilstm-crf.* Frontiers of Information Technology & Electronic Engineering, 20(6):872–884, 2019. 9

[50] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton: *Imagenet classification with deep convolutional neural networks.* Communications of the ACM, 60(6):84–90, 2017. 9

[51] Simonyan, Karen and Andrew Zisserman: *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014. 9, 27, 63, 64, 77

[52] Ioffe, Sergey and Christian Szegedy: *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* In *International conference on machine learning*, pages 448–456. PMLR, 2015. 10

[53] Huisman, Mike, Jan N Van Rijn, and Aske Plaat: *A survey of deep meta-learning.* Artificial Intelligence Review, 54(6):4483–4541, 2021. 11, 18

[54] Thrun, Sebastian and Lorien Pratt: *Learning to learn.* Springer Science & Business Media, 2012. 11

[55] Thrun, Sebastian: *Lifelong learning algorithms.* In *Learning to learn*, pages 181–209. Springer, 1998. 11

[56] Hochreiter, Sepp, A Steven Younger, and Peter R Conwell: *Learning to learn using gradient descent.* In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001. 11

[57] Nichol, Alex, Joshua Achiam, and John Schulman: *On first-order meta-learning algorithms.* arXiv preprint arXiv:1803.02999, 2018. 11, 14, 29, 37, 39

[58] Finn, Chelsea, Pieter Abbeel, and Sergey Levine: *Model-agnostic meta-learning for fast adaptation of deep networks.* In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 11, 65

[59] Santoro, Adam, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap: *Meta-learning with memory-augmented neural networks.* In *International conference on machine learning*, pages 1842–1850. PMLR, 2016. 11

[60] Andrychowicz, Marcin, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas: *Learning to learn by gradient descent by gradient descent.* arXiv preprint arXiv:1606.04474, 2016. 11, 14

[61] Schmidhuber, Jürgen: *A neural network that embeds its own meta-levels.* In *IEEE International Conference on Neural Networks*, pages 407–412. IEEE, 1993. 11

[62] Li, Weigang: *A study of parallel self-organizing map.* Transactions of Nonferrous Metals Society of China English Edition, 9(1):27–35, 1998. 12

[63] Miller, Erik G, Nicholas E Matsakis, and Paul A Viola: *Learning from one example through shared densities on transforms.* In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000. 12

[64] Lake, Brenden, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum: *One shot learning of simple visual concepts.* In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011. 12

[65] Chopra, Sumit, Raia Hadsell, and Yann LeCun: *Learning a similarity metric discriminatively, with application to face verification.* In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 12

[66] Vinyals, Oriol, Charles Blundell, Timothy Lillicrap, Daan Wierstra, *et al.*: *Matching networks for one shot learning.* In *Advances in neural information processing systems*, pages 3630–3638, 2016. 12, 65

[67] Neculoiu, Paul, Maarten Versteegh, and Mihai Rotaru: *Learning text similarity with siamese recurrent networks.* In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157, 2016. 12

[68] Banerjee, Arindam, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty: *Clustering with bregman divergences.* Journal of machine learning research, 6(10), 2005. 13

[69] Baltrušaitis, Tadas, Chaitanya Ahuja, and Louis Philippe Morency: *Multimodal machine learning: A survey and taxonomy.* IEEE transactions on pattern analysis and machine intelligence, 41(2):423–443, 2018. 15, 16, 24

[70] Wang, Wenshan, Pengfei Liu, Su Yang, and Weishan Zhang: *Dynamic interaction networks for image-text multimodal learning.* Neurocomputing, 379:262–272, 2020. 16, 17, 21, 22, 23, 24, 25

[71] Joshi, Gargi, Rahee Walambe, and Ketan Kotecha: *A review on explainability in multimodal deep neural nets.* IEEE Access, 9:59800–59821, 2021. 16, 17, 89

[72] Cimtay, Yucel, Erhan Ekmekcioglu, and Seyma Caglar-Ozhan: *Cross-subject multimodal emotion recognition based on hybrid fusion.* IEEE Access, 8:168865–168878, 2020. 16, 17

[73] Eloff, Ryan, Herman A Engelbrecht, and Herman Kamper: *Multimodal one-shot learning of speech and images.* In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8623–8627. IEEE, 2019. 17, 21, 22, 23, 25

[74] Passalis, Nikolaos, Alexandros Iosifidis, Moncef Gabbouj, and Anastasios Tefas: *Robust hypersphere-based weight imprinting for few-shot learning.* In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1392–1396. IEEE, 2021. 21, 22, 23, 27, 28

[75] Islam, Md Tamzeed and Shahriar Nirjon: *Soundsemantics: exploiting semantic knowledge in text for embedded acoustic event classification.* In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 217–228, 2019. 21, 22, 23, 26, 27

[76] Yu, Yunlong, Zhong Ji, Jungong Han, and Zhongfei Zhang: *Episode-based prototype generating network for zero-shot learning.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14035–14044, 2020. 21, 22, 23, 27, 28, 44, 66

[77] Li, Xinpeng, Dan Zhang, Mao Ye, Xue Li, Qiang Dou, and Qiao Lv: *Bidirectional generative transductive zero-shot learning.* Neural Computing and Applications, 33(10):5313–5326, 2021. 21, 22, 23, 26, 27

[78] Ji, Zhong, Zhishen Hou, Xiyao Liu, Yanwei Pang, and Jungong Han: *Information symmetry matters: A modal-alternating propagation network for few-shot learning.* IEEE Transactions on Image Processing, 31:1520–1531, 2022. 21, 23, 25, 28, 65, 77, 87

[79] Li, Mingxi, Ronggui Wang, Juan Yang, Lixia Xue, and Min Hu: *Multi-domain few-shot image recognition with knowledge transfer.* Neurocomputing, 442:64–72, 2021. 21, 22, 23, 26, 65

[80] Fang, Zhiyu, Xiaobin Zhu, Chun Yang, Zheng Han, Jingyan Qin, and Xu Cheng Yin: *Learning aligned cross-modal representation for generalized zero-shot classification.* In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6605–6613, 2022. 21, 22, 23, 26, 27, 66

[81] Zhao, Jiabao, Xin Lin, Yifan Yang, Jing Yang, and Liang He: *Cross-modal knowledge distillation for fine-grained one-shot classification.* In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4295–4299. IEEE, 2021. 21, 22, 23, 24, 25, 28, 44, 65

[82] Tonge, Ashwini and Cornelia Caragea: *Dynamic deep multi-modal fusion for image privacy prediction.* In *The World Wide Web Conference*, pages 1829–1840, 2019. 21

[83] Bendre, Nihar, Kevin Desai, and Peyman Najafirad: *Generalized zero-shot learning using multimodal variational auto-encoder with semantic concepts.* In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1284–1288. IEEE, 2021. 21, 22, 23, 26, 27, 44, 66

[84] Ding, Kaize, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu: *Graph prototypical networks for few-shot learning on attributed networks.* In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 295–304, 2020. 21, 22, 23, 25

[85] Pan, Chongyu, Jian Huang, Jianguo Hao, and Jianxing Gong: *Towards zero-shot learning generalization via a cosine distance loss.* Neurocomputing, 381:167–176, 2020. 21, 22, 23, 27, 28

[86] Jia, Ye, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu: *Leveraging weakly supervised data to improve end-to-end speech-to-text translation.* In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184. IEEE, 2019. 22

[87] Scarselli, Franco, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini: *The graph neural network model.* IEEE transactions on neural networks, 20(1):61–80, 2008. 25

[88] Zhu, Jun Yan, Taesung Park, Phillip Isola, and Alexei A Efros: *Unpaired image-to-image translation using cycle-consistent adversarial networks.* In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 26

[89] Xian, Yongqin, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata: *Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(9):2251–2265, 2019. 26

[90] Kingma, Diederik P and Max Welling: *Auto-encoding variational bayes.* arXiv preprint arXiv:1312.6114, 2013. 26

[91] Schmitz, Matheus, Roger Immich, Gustavo Pessin, and Geraldo Pereira Rocha Filho: *Towards the categorization of brazilian financial market headlines.* IEEE Latin America Transactions, 20(2):344–351, 2021. 33

[92] Schulte, Johannes P, Felipe T Giuntini, Renato A Nobre, Khalil C do Nascimento, Rodolfo I Meneguette, Weigang Li, Vinícius P Gonçalves, and Geraldo P Rocha Filho: *Elinac: Autoencoder approach for electronic invoices data clustering.* Applied Sciences, 12(6):3008, 2022. 33

[93] Kemény, Ferenc and Beat Meier: *Multimodal sequence learning.* Acta psychologica, 164:27–33, 2016. 33

[94] Rajaraman, Sivaramakrishnan, Sameer K Antani, Mahdieh Poostchi, Kamolrat Silamut, Md A Hossain, Richard J Maude, Stefan Jaeger, and George R Thoma: *Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images.* PeerJ, 6:e4568, 2018. 41, 44

[95] Wah, Catherine, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie: *The caltech-ucsd birds-200-2011 dataset.* 2011. `https://authors.library.caltech.edu/27452/`. 41, 44

[96] Nilsback, Maria Elena and Andrew Zisserman: *Automated flower classification over a large number of classes.* In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 41, 44

[97] Reed, Scott, Zeynep Akata, Honglak Lee, and Bernt Schiele: *Learning deep representations of fine-grained visual descriptions.* In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016. 41, 44

[98] Zhang, Xiang and Yann LeCun: *Which encoding is the best for text classification in chinese, english, japanese and korean?* arXiv preprint arXiv:1708.02657, 2017. 54

[99] Devlin, Jacob, Ming Wei Chang, Kenton Lee, and Kristina Toutanova: *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018. 61, 77, 78, 80, 87

[100] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, *et al.*: *Imagenet large scale visual recognition challenge.* International journal of computer vision, 115(3):211–252, 2015. 64

[101] Chen, Zhikui, Xu Zhang, Wei Huang, Jing Gao, and Suhua Zhang: *Cross modal few-shot contextual transfer for heterogenous image classification.* Frontiers in Neurorobotics, 15:56, 2021. 65

[102] Xu, Rui, Lei Xing, Shuai Shao, Lifei Zhao, Baodi Liu, Weifeng Liu, and Yicong Zhou: *Gct: Graph co-training for semi-supervised few-shot learning.* IEEE Transactions on Circuits and Systems for Video Technology, 2022. 65

[103] Bateni, Peyman, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal: *Improved few-shot visual classification.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14493–14502, 2020. 65

[104] Guang, Jinzheng and Jianru Liang: *Cmsea: Compound model scaling with efficient attention for fine-grained image classification.* IEEE Access, 10:18222–18232, 2022. 66

[105] Chen, Jianpin, Heng Li, Junlin Liang, Xiaofan Su, Zhenzhen Zhai, and Xinyu Chai: *Attention-based cropping and erasing learning with coarse-to-fine refinement for fine-grained visual classification.* Neurocomputing, 2022. 66

[106] Liu, Xinda, Lili Wang, and Xiaoguang Han: *Transformer with peak suppression and knowledge guidance for fine-grained image recognition.* Neurocomputing, 492:137–149, 2022. 66

[107] Chen, Xingyu, Jin Li, Xuguang Lan, and Nanning Zheng: *Generalized zero-shot learning via multi-modal aggregated posterior aligning neural network.* IEEE Transactions on Multimedia, 2020. 66

[108] Dodge, Jesse, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A Smith, Nicole DeCario, and Will Buchanan: *Measuring the carbon intensity of ai in cloud instances.* In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1877–1894, 2022. 76

[109] Wu, Carole Jean, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, *et al.*: *Sustainable ai: Environmental implications, challenges and opportunities*. Proceedings of Machine Learning and Systems, 4:795–813, 2022. 76, 84

[110] García-Martín, Eva, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn: *Estimation of energy consumption in machine learning.* Journal of Parallel and Distributed Computing, 134:75–88, 2019. 77