Universidade Estadual de Campinas
Instituto de Computação

Thamiris Florindo Coelho

Transformers-Based Few-Shot Learning for Scene Classification in Child Sexual Abuse Imagery

Aprendizado por Poucas Amostras Baseado em Transformers para Classificação de Cenas em Imagens de Abuso Sexual Infantil

CAMPINAS
2023

Thamiris Florindo Coelho

Transformers-Based Few-Shot Learning for Scene Classification in Child Sexual Abuse Imagery

Aprendizado por Poucas Amostras Baseado em Transformers para Classificação de Cenas em Imagens de Abuso Sexual Infantil

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestra em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientadora: Profa. Dra. Sandra Eliza Fontes de Avila
Co-supervisor/Coorientador: Prof. Dr. Jefersson Alex dos Santos

Este exemplar corresponde à versão final da Dissertação defendida por Thamiris Florindo Coelho e orientada pela Profa. Dra. Sandra Eliza Fontes de Avila.

CAMPINAS
2023

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

C65t

Coelho, Thamiris Florindo, 1997-
Transformers-based few-shot learning for scene classification in child abuse imagery / Thamiris Florindo Coelho. – Campinas, SP : [s.n.], 2023.

Orientador: Sandra Eliza Fontes de Avila.
Coorientador: Jefersson Alex dos Santos.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina. 2. Aprendizado profundo. 3. Crime sexual contra as crianças. 4. Visão por computador. 5. Classificação de imagem. 6. Transformer (Arquitetura de computador). I. Avila, Sandra Eliza Fontes de, 1982-. II. Santos, Jefersson Alex dos, 1984-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações Complementares

Título em outro idioma: Aprendizado por poucas amostras baseado em Transformers para classificação de cenas em imagens de abuso sexual infantil
Palavras-chave em inglês:
Machine learning
Deep learning
Childhood sexual abuse
Computer vision
Image classification
Transformer (Computer architecture)
Área de concentração: Ciência da Computação
Titulação: Mestra em Ciência da Computação
Banca examinadora:
Sandra Eliza Fontes de Avila [Orientador]
Moacir Antonelli Ponti
Esther Luna Colombini
Data de defesa: 10-11-2023
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0009-0001-4894-0347
- Currículo Lattes do autor: http://lattes.cnpq.br/3046744603908167

Universidade Estadual de Campinas
Instituto de Computação

Thamiris Florindo Coelho

Transformers-Based Few-Shot Learning for Scene Classification in Child Sexual Abuse Imagery

Aprendizado por Poucas Amostras Baseado em Transformers para Classificação de Cenas em Imagens de Abuso Sexual Infantil

Banca Examinadora:

- Profa. Dra. Sandra Eliza Fontes de Avila (Orientadora)
  Universidade Estadual de Campinas
- Prof. Dr. Moacir Antonelli Ponti
  Universidade de São Paulo
- Profa. Dra. Esther Luna Colombini
  Universidade Estadual de Campinas

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 10 de novembro de 2023

Agradecimentos

Gostaria de começar dizendo que a trajetória para chegar até aqui não foi nada fácil. Estudar na Unicamp sempre foi o meu sonho, mas vindo de uma escola pública parecia impossível. Entrar na universidade foi incrível, mas se manter nela até o fim, foi muito desafiador. Muitas vezes eu me sentia muito atrasada em relação aos meus colegas de turma, e durante a faculdade eu pensava que conseguir meu diploma do bacharelado seria o máximo. Ter entrado no Mestrado e estar hoje defendendo minha dissertação é inimaginável e eu tenho muito orgulho de ter chegado até aqui. Mas seria impossível chegar até aqui sozinha, então gostaria de agradecer algumas pessoas que foram muito importantes nessa trajetória.

Quero começar agradecendo minha orientadora Profa. Dra. Sandra Eliza Fontes de Avila, que aceitou me orientar e viu um potencial em mim que nem eu mesma via. Nossa trajetória começou muito antes do mestrado, em um trabalho de iniciação científica, que no fim percebemos que não era do interesse de nenhuma das duas. Mas depois disso, todas às vezes que nos encontrávamos nos corredores, a Sandra sempre me instigava a fazer um mestrado. Obrigada por além de ser uma orientadora excepcional, acompanhando sempre de perto o meu progresso e participando ativamente da pesquisa, ser uma pessoa que se preocupa com o meu desenvolvimento pessoal. Com toda certeza me tornei uma pessoa melhor após conviver e aprender tando com a Sandra.

Gostaria também de agradecer ao meu coorientador Prof. Dr. Jefersson Alex dos Santos e as pessoas que fazem ou já fizeram parte do nosso grupo de pesquisa Andreza, Camila, Gil, João, Leo e Pedro. Vocês foram fundamentais no desenvolvimento dessa dissertação, me ajudando com ideias para o desenvolvimento do projeto, além de serem inspirações para mim. Em especial, gostaria de agradecer a Leo Sampaio, que chegou mais no fim do mestrado, mas que me acompanhou bem de perto, me auxiliando muito para que o resultado da dissertação fosse o apresentado. Sem ela essa dissertação não seria a mesma.

Além das pessoas que ajudaram na execução do trabalho, gostaria de agradecer a minha família, Marcus, Rose e Lucca por me darem muito amor e suporte durante esse tempo, se não fosse meu pai me pedindo muito encarecidamente para realizar o mestrado, não sei se teria chegado até aqui. Agradeço também ao meu namorado, Lucas, que foi meu porto seguro durante todo o processo de altos e baixos que foi realizar esse Mestrado, obrigada por acreditar em mim e me apoiar não só com palavras, mas com ações. Por mais que elas não saibam ler essa dedicatória, agradeço as minhas gatas Atena, Aurora e Artemis (que infelizmente não está mais nesse plano), que foram parte do meu suporte emocional durante esse período. Agradeço também aos meus amigos, que durante todo o processo, tiveram que ouvir minhas reclamações sobre como pesquisar pode ser muito difícil e como eu estava sofrendo nesse processo, obrigada Beatriz, Daniela, Isabela, Mariana, Ruan e Tainá.

Por fim, agradeço ao Santander e ao Instituto de Computação da Unicamp, que, através da bolsa Alumini, financiaram essa pesquisa.

Resumo

O abuso sexual é um crime que afeta muitas crianças ao redor do mundo. Só no último ano foram feitas mais de 32 milhões de denúncias de abuso sexual infantil foram feitas para o Centro Nacional para Crianças Desaparecidas e Exploradas. Infelizmente, o volume de material multimídia produzido diariamente é muito maior do que a capacidade de análise visual feita por profissionais da aplicação da lei. Nesse contexto, ter uma ferramenta confiável que classifique automaticamente o material de abuso sexual infantil é essencial. Métodos de Aprendizado Profundo, estado da arte para problemas de classificação de imagem, demandam grandes quantidades de dados para treinamento. Mesmo com um grande volume de dados disponíveis, a anotação dessas imagens é muito custosa. Além disso, devido a barreiras legais e éticas, esses dados sensíveis só podem ser acessados por agentes da polícia. Para lidar com isso, alguns métodos tentam ajudar as investigações resolvendo um problema computacional relacionado. A classificação de cenas internas pode ajudar a detectar ambientes nos quais esse tipo de conteúdo é tipicamente composto. No entanto, nesses ambientes, a presença de um objeto pode mudar completamente a classificação. Módulos de atenção da arquitetura Transformers podem ajudar o modelo a focar nas partes essenciais dos dados para resolver algumas tarefas. Assim, para focar em objetos presentes em cenas, esse trabalho utiliza modelos baseados em Transformers. Além disso, para lidar com o desafio de anotar os dados, utilizamos aprendizagem com poucas amostras (*few-shot learning*), uma técnica de aprendizado supervisionado que aprende utilizando poucas amostras anotadas. Nesta dissertação de mestrado analisamos alguns modelos de *few-shot learning* clássicos e comparamos modelos baseados em Transformers para classificação de cenas de ambientes internos. Observamos também que a maioria dos trabalhos analisados utiliza um mesmo método de agrupamento de vetores de características, portanto, nesse sentido investigamos o uso de diferentes métodos, concluindo que agregar os vetores utilizando a média é o melhor para o conjunto de cenas internas. Nossos resultados indicam que o uso de Transformers é benéfico no contexto de classificação de cenas internas. Além disso, para o conjunto de dados de cenas internas utilizado, utilizar a média para agregar os vetores de características levou aos melhores resultados, provavelmente porque no nosso contexto a média foi uma boa representação. Nosso modelo final atingiu 73,50 ± 0,09% de acurácia com 95% de confiança na tarefa de classificação de cenas internas utilizando apenas 5 amostras anotadas por classe para a classificação. Em cooperação com especialistas da Polícia Federal Brasileira pudemos avaliar nosso modelo em um conjunto de dados de abuso sexual infantil anotado para cenas internas, nosso modelo atingiu uma acurácia balanceada com 95% de confiança de 63,38 ± 0,09%, avaliamos que os resultados foram promissores, indicando que a utilização da técnica proposta pode auxiliar em um processo de triagem.

Abstract

Sexual abuse is a crime that affects many children around the world. In just the past year, more than 32 million reports of child sexual abuse were made to the National Center for Missing & Exploited Children. Unfortunately, the volume of multimedia material produced daily is much greater than the visual analysis capacity of law enforcement agents. In this context, having a reliable tool that can automatically classify child sexual abuse material is essential. Deep learning methods, state-of-the-art for image classification problems, require large amounts of data for training. Even with a large volume of available data, annotating these images is very costly. Additionally, law enforcement agents can only access this sensitive data due to legal and ethical barriers. To address this, some methods try to assist investigations by solving a related computational problem. The classification of indoor scenes can help detect environments where this type of content is typically found. However, in these environments, the presence of an object can completely change the classification. Attention modules of the Transformer architecture can help the model focus on the essential parts of the data to solve some tasks. Thus, this work utilizes Transformer-based models to focus on objects present in scenes. Also, to address the challenge of annotating data, we use few-shot learning, a supervised learning technique that learns using a few annotated samples. In this Master's thesis, we analyze some classic few-shot learning models and compare Transformer-based models for the classification of indoor scenes. We also observe that most of the analyzed works use the same method for aggregating feature vectors; therefore, in this regard, we investigate using different methods, concluding that aggregating vectors using the mean is the best for the set of indoor scenes. Our results indicate that using Transformers is beneficial in indoor scene classification. Furthermore, for the dataset of indoor scenes used, using the mean to aggregate feature vectors led to the best results, probably because, in our context, the mean was a good representation. Our final model achieved an accuracy of $73.50 \pm 0.09\%$ with 95% confidence in classifying indoor scenes using only 5 annotated samples per class for classification. In cooperation with experts from the Brazilian Federal Police, we evaluated our model on a dataset of annotated child sexual abuse for indoor scenes, and our model achieved a balanced accuracy of $63.38 \pm 0.09\%$ with 95% confidence. We believe the results were promising, indicating that the proposed technique can assist in screening.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Child sexual abuse is a crime that affects about 9–19.7% of girls and 3–7.9% of boys [61, 82, 102], including indecent exposure, forced sex, and sex trafficking. Even after escaping from abusers, victims face immediate and lasting consequences, such as an increased risk of depression, substance abuse, and suicide [84]. Over the past decade, online sharing platforms have been used to leverage the distribution of child sexual abuse images and videos [13].

According to the USA's National Center for Missing & Exploited Children (NCMEC) [1], in 2022, the number of reports of suspected child sexual exploitation was more than 32 million. These reports included more than 87 million image/video files containing child sexual abuse. From 2020 to 2022, there was an increase of more than 33% in the number of Child Sexual Abuse Material (CSAM), verified by NCMEC's CyberTipline (the USA's centralized reporting system for the online exploitation of children). From the images and videos analyzed last year, more than 70 million were unique when compared to hash sources [50].

From 2012 to 2015, according to data from the National Criminalistics Management System of the Federal Police of Brazil, forensic examinations were carried out on over 4,000 computer storage devices, surpassing 500 TB of analyzed data [112]. Unfortunately, on a day-to-day basis, it turns out that the volume of material produced is much greater than the capacity for the visual analysis carried out by law enforcement professionals. In this context, intelligent and continuous automatic classification of child sexual abuse material is paramount.

Due to legal and ethical barriers, such sensitive data cannot be accessed by anyone aside from the police force. For this reason, most popular child sexual abuse detectors use hash comparison [10, 23, 81]. Microsoft's PhotoDNA [50] is the best-known tool based on hash comparison, and big companies like Meta, X (formerly known as Twitter), and Google use it. Even though this tool is key to identifying the reappearance of existing CSAM, those methods are sensitive to minor modifications on the visual content, such as scaling or color changing [97], in a way that the method can not match the hash of new content with existing ones. Therefore, more robust techniques like Deep Neural Networks started to be used to classify CSAM [12, 71, 111, 115].

To deal with the inability to access the data, some methods try to help classify CSAM without classifying sensitive data but solve a related problem. Anda et al. [4], Gangwar et al. [34], Macedo et al. [71], for example, attempted this problem via automatic nudity detection and age estimation. The reason is that Machine Learning models can be trained using external data combining age estimation and nudity datasets and then use those models to help in the

CSAM detection.

There are other ways to help CSAM investigation; scene classification, for example, can also be helpful, detecting environments in which this type of content is typically recorded or photographed [13]. Nonetheless, this approach is yet an underexplored dimension of CSAM investigation and could be a great strategy to triage possible CSAM. From this perspective, we investigate the triage of CSAM, classifying indoor scenes, since those contents are mostly recorded on those environments [57].

Deep Learning methods are state-of-the-art to solve many problems, particularly image and video classification problems [51, 74, 83]. However, the best methods demand massive amounts of annotated data to achieve good results. Even if the methods to classify such sensitive data run inside the police-restricted environments, annotating this kind of data is challenging, and being exposed to it for an extended period can compromise mental well-being [55].

Therefore, to tackle both problems – the lack of access to data and the challenge of annotating the data, we developed an indoor scene classifier through supervised learning techniques from a few annotated samples (few-shot learning) [8, 114]. This line of research has attracted attention since it answers the need to obtain models that generalize well for applications where access to large sets of annotated images is impractical or costly. That is the case with the classification of CSAM.

1.1 Problem Description

Artificial Intelligence (AI) methods have achieved state-of-the-art results on several tasks. In many fields, AI can even perform better than humans. For example, residual networks [42] (ResNets) obtained better performance than humans classifying ImageNet [24]. Despite yielding impressive results, these methods heavily depend on training with large-scale datasets and often struggle to generalize when only a limited number of samples are accessible. In contrast, Few-Shot Learning (FSL) [30, 31] was proposed to teach machines from a few samples, at most 20 samples per class. Those methods are helpful when annotated examples, such as medical applications, face recognition, and CSAM classification, are hard or impossible to acquire. Many applications can apply FSL, such as image classification, object tracking, and video event detection.

In this Master's thesis, we use few-shot learning for indoor scene classification since most visual material for child sexual abuse is recorded in those environments. Although there are several datasets for scene recognition, and this kind of data is easy to annotate, FSL is still relevant to our problem since no FSL work is proposed for this task. Also, our goal is to apply the method to help CSAM investigation through the triage of possible CSAM based on the scene. In this scenario, only a few samples are annotated, especially using scene labels. This way, police agents will rapidly adjust the model using only a few annotated samples, considering their data. As far as we know, scene recognition has yet to be explored by FSL methods.

In most works, outdoor and indoor classifications are considered together. However, these environments have very different characteristics. Indoor environments are more complex; the objects presented in the scene usually define the kind of room. For example, a room with a bed and a desk is probably a bedroom, but if we remove the bed, it is probably an office [85]. In

comparison, global information is more critical in outdoor scenes, and both local and global information matters in indoor scenes. These vast differences can make models with impressive results on outdoor scenes perform poorly in indoor scenes [125].

With that in mind, the models must focus on the objects in the scene to classify them correctly. Transformers-based models [108] can give attention to the most essential parts of the data to solve some tasks. Figure 1.1 shows some examples of attention maps focusing on objects or people in an image. In addition, the technique recently outperformed ResNets results for ImageNet classification [28]. In this work, we combine Transformer-based techniques with few-shot learning.

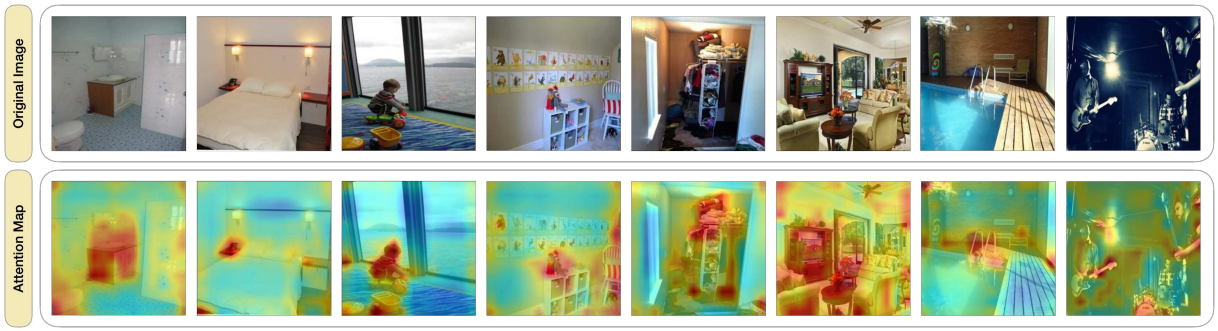


Figure 1.1: Attention maps from the final model of this Master's thesis. The patches of the image with higher attention are represented in red, while the lower attention is represented in blue.

1.2 Motivations and Challenges

It is easy for humans to identify sexually explicit material ("I know it when I see it"[1]), but CSAM goes beyond that. According to Gillespie [35], all indecent content containing children may be classified as CSAM, but it is difficult to determine whether an image is indecent or not. Sometimes, it is difficult to determine the person's age, mostly pre-teens and teenagers; in that case, some children can appear older due to physical characteristics or clothes. Even though it is easier to identify images or videos containing explicit sexual acts, exposure to such material can compromise agents' mental health [55]. Therefore, the smaller the number of CSAM an agent needs to analyze, the better.

Therefore, the automatic detection of CSAM is crucial, and it can support police agents to filter and review the material. However, in addition to the difficulty of defining people's ages and identifying the presence of indecency in the images, access to this extremely sensitive content is illegal. Researchers must develop solutions that help police agents detect this material without accessing such data. Bursztein et al. [13] argue that clustering scenes and recognizing objects and landmarks could help investigations.

This work proposes using few-shot learning strategies to build generalized models that can adapt to new tasks using only a little data. Traditionally, Deep Learning-based models need a large amount of data to perform well; thus, working with only a few data is challenging.

[1]The phrase was used in 1964 by United States Supreme Court Justice Potter Stewart to describe his threshold test for obscenity in Jacobellis vs. Ohio.

Beyond that, we are tackling a very complex problem, and our model needs to have the ability to recognize indoor environments in such sensitive data. To that, we focus on metric-learning embedding-based few-shot learning [47, 101, 104, 122], an FSL category that works with non-parametric learning, and this characteristic is something needed for CSAM, our final task.

Additionally, most metric-learning embedding-based few-shot works use the average to aggregate the feature vectors extracted from their backbones without comparing this aggregator with others. Improving this step would, in turn, improve the CSAM application. For that reason, we want to understand the impact of the aggregation method in our work; therefore, we investigate which aggregator(e.g., average pooling [58], max pooling [11], self-attention [94]) is the best for few-shot indoor classification.

1.3 Objectives

The main objective of this Master's thesis is to use little data to train a model to classify indoor scene images. We can summarize our objectives:

O1. To build a classification model for indoor scenes using only a few labeled samples.

O2. To compare the performance of classic few-shot learning models with Transformer-based few-shot learning in indoor scene classification tasks.

O3. To support the CSAM investigation process with a reliable classifier.[2]

1.4 Research Questions

Considering few-shot learning scenarios, the research questions this project aims to answer are:

Q1. Do Transformers-based methods outperform convolutional neural network methods for indoor scenes' classification?

Q2. What is the more accurate way to use Transformers for indoor classification? As a feature adapter or a feature extractor?

Q3. What are the impacts of feature vector aggregators on few-shot indoor classification models?

Q4. How to develop an indoor classification model that generalizes for the CSAM environment?

[2]Some members of our research group are agents of the Brazilian Federal Police who are pursuing Master's or Ph.D. degrees, and they can evaluate our methods in a CSAM environment.

1.5 Contributions

This Master's thesis is the first work that uses few-shot learning methods to classify indoor scenes. We compared few-shot Transformer-based and purely convolutional approaches and showed that Transformer-based approaches lead to better results for the target task.

We also compared different aggregation methods to understand the best way to aggregate feature vectors of samples from the same class. The average – the most used method in few-shot learning – is actually the best way to aggregate features for our target task. As far as we know, this is the first work that compares aggregation methods in few-shot learning.

Furthermore, we present the results of our final model on a CSAM dataset. For the classification of indoor environments on the CSAM task, the model achieved a balanced accuracy of 63.38 ± 0.09% for the few-shot evaluation pipeline, using only five samples per class to classify the samples. Our results show that indoor scene features are relevant in the CSAM classification.

1.6 Outline

We organized the remainder of this work as follows. In Chapter 2, we review the related concepts of few-shot learning classification, aggregation methods, transductive and inductive learning, and Transformers. In Chapter 3, we overview the literature on scene classification and embedding learning, highlighting the usage of Transformers in few-shot learning. In Chapter 4, we describe the datasets used for training and testing following the proposed methodology. In Chapter 5, we introduce our methodology and how we train and validate the models. In Chapter 6, we analyze the results of our experiments. Finally, in Chapter 7, we conclude our work, presenting the main findings and future works.

Chapter 2

Related Concepts

This chapter presents the concepts relevant to understanding this Master's thesis. In Section 2.1, we explain the concept of meta-learning, the techniques of this concept, and lastly, how it is used in few-shot learning. In Section 2.2, we present the concept of few-shot learning and its methods. In Section 2.2.1, we overview the aggregation methods since in many metric-based few-shot learning approaches, which will be explained in the following sections, they aggregate feature vectors to generate a unique vector per class; in this work, we investigate what is the best function to aggregate those vectors. Few-shot learning can be tackled from two different learning paradigms: inductive or transductive. So, in Section 2.2.2, we explain the difference between those paradigms and why we chose inductive for our exploration. Besides that, we focus on a specific FSL challenge, using Transformers in the learning process. Then, in Section 2.3, we explain how Transformers works and its particularities. We aim to develop a metric-based few-shot learning method trained inductively using Transformers techniques, such as self-attention.

2.1 Meta-Learning

Unlike most Machine Learning methods, when learning a new task, humans often use the knowledge acquired from prior experience—considering the vast background that humans have, it becomes easier to learn new tasks [56]. So, learning from previous experience can also benefit machine learning models in learning a new task faster. From that perspective, Meta-Learning proposes learning from prior knowledge, *learning how to learn* and then applying this knowledge to new tasks.

Finding one exact definition of meta-learning is difficult since different views of this paradigm can be found in the literature [46]. For Thrun and Pratt [105], meta-learning is characterized by an algorithm that improves performance with more training experience and several tasks. For a more contemporary definition, meta-learning is learning an algorithm over several learning episodes [46]. We follow the contemporary view of meta-learning.

In this contemporary view, meta-learning is a learning algorithm that can generalize across tasks and can learn new tasks faster. For that in the meta-training phase we have $D_{meta-test}$, a dataset composed of N tasks, described as $D_{meta-train} = \{(D^{train}_{meta-train}, D^{val}_{meta-train})^{(i)}\}^{N}_{i=1}$, where D are different tasks, and each task is defined by $D_i = ((x^i_1, y^i_1), (x^i_2, y^i_2), \ldots, (x^i_k, y^i_k))$.

From that, we want to learn meta-parameters that describe the meta-learning phase so that the meta-learning problem can be defined as:

$$\theta^* = \arg\max_{\theta} \log \phi(\theta | D_{meta-train}) \tag{2.1}$$

Now, in the meta-testing phase, $D_{meta-test} = \{(D^{train}_{meta-test}, D^{test}_{meta-test})^{(i)}\}^{Q}_{i=1}$ where Q is the number of target tasks. In this stage, we use the learned meta-knowledge θ^* to train the model on the unseen tasks of $D_{meta-test}$, defined in Equation 2.2. Then we can evaluate the ω^* on $D^{test}_{meta-test}$.

$$\omega^* = \arg\max_{\omega} \log \phi(\omega | \theta, D^{train}_{meta-test}) \tag{2.2}$$

The learned θ^* could be an optimization strategy, a learning model, or even the initial parameters of a learning model. From that, Yao et al. [121] categorize meta-learning in three main research lines: (1) optimization-based, (2) model-based (black box), and (3) metric-learning.

Optimization-based methods aim to learn an optimized initialization for the model parameters, such that the model could adapt to new tasks with only a few steps of the gradient [32, 60, 93].

Model-based (black-box) methods do not generate sample classification probabilities. Instead, the idea of those models is to update the network parameters fast [88, 95].

Metric-learning methods combine parametric and non-parametric learning in the meta-training and meta-testing stage; the objective is to learn a distance metric that better generalizes for new tasks [101, 104, 110].

There are several applications of meta-learning, some examples are: few-shot learning [32, 110], multi-task learning [33, 105], and meta-reinforcement [64, 92] learning. In this Master's thesis, we focus on FSL, where most works can be categorized as optimization-based or metric-learning. We find it more interesting to investigate FSL methods based on metric learning.

2.2 Few-Shot Learning Classification

In traditional Machine Learning techniques, the models learn how to classify images using large annotated datasets. However, for some real-world applications, this is unfeasible. FSL recently gained attention in this scenario, as it aims to solve a task using a minimal number of labeled examples. This technique aims to learn a good classifier given only limited samples of each class; typically, the number of samples per class is between 0 and 20.

The idea behind FSL is to use prior knowledge to teach the model how to improve on the target task. We usually have three sets [19, 27] to do that. The first set is called *base set*, a large dataset containing multiple classes to pre-train the model. The second set is the *support set* (also called *novel set*), which contains samples from the FSL task; the classes in this set and the base set are disjointed. The last set is the *query set*, which contains the samples we want to predict.

The support set comprises K samples per class and N different classes. Based on that, an FSL task can also be called N-way K-shot learning. There are two particular cases; first, when the number of labeled samples per class is zero ($K = 0$), it is called *zero-shot learning*, and when $K = 1$, it is called *one-shot learning*. Two main approaches are used to solve FSL problems: transfer learning and meta-learning.

In solutions based on *transfer learning*, the idea is to train a feature extractor using the base set to extract the features from the novel set. On top of the feature extractor, many approaches can be taken to compare and classify the samples in the query set into the classes contained in the support set. In this approach, the performance is related to the similarity between the datasets [5, 63, 69].

In contrast, the model gradually learns generic information from the base set in solutions based on *meta-learning* at training time. Then, at test time, the meta-learner for the FSL task is generalized using the support set. To learn gradually, the training process is made by episodes. Each episode is similar to the few-shot task, which contains K samples of N classes randomly chosen from the base set and the same number of the query set in the validation set [101, 104, 110]. Figure 2.1 shows the separation of each of the used datasets, highlighting the episode on episodic training.

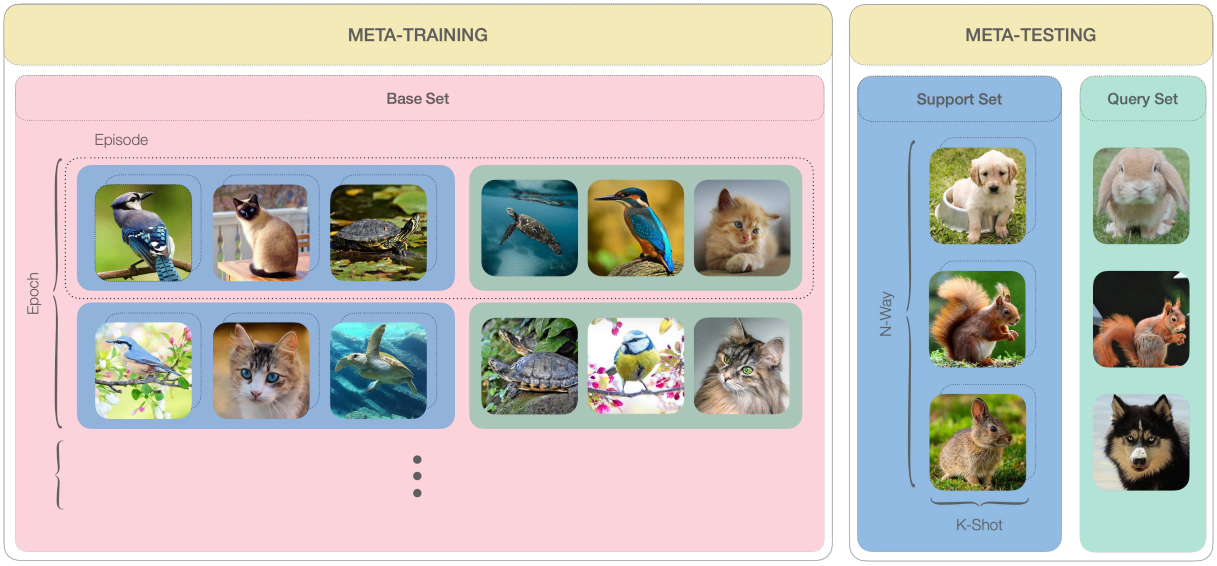


Figure 2.1: Episodic training. Figure adapted from Bennequin [9].

The base set must be labeled to mimic the support and query sets during the meta-training stage. Some works also have a pre-training stage, following a self-supervised approach [44, 62]. However, in those cases, it is crucial to guarantee that none of the sets used in pre-meta-testing have intersectional classes.

General machine learning methods in the testing stage classify unseen samples from the same classes seen during training based on what was learned from the training set. In contrast, in meta-testing, the objective is to classify classes never seen during meta-training. So, during meta-training, the model is trained using the base set, and then, on meta-testing, it uses the support set to classify the samples in the query set.

Besides the training approaches, FSL can be classified into three categories [8]: optimization-based, data-based, or metric-based. We explain each of these ways in the following.

Optimization-Based

This method searches for the best optimization to have an initialization that better generalizes the model. To learn these optimizations, first, it learns the parameters from other tasks (base set) and then uses the support set to refine the parameters [114]. As the parameters are optimized for the target task using the support set in meta-testing time, this method can sacrifice speed for precision.

Data-Based

This method uses augmentation to increase the number of samples available in the support set. Then, with a large train set, it is possible to use standard Machine Learning algorithms. This method is usually applied as a pre-processing step in FSL. It is possible to use hand-crafted techniques, such as flipping, rotation, cropping, or more advanced ones.

There are three main ways to use advanced techniques to augment the data set. First, we can transform samples from the train set, transforming each sample into several samples with variation [41, 98]. Another way to augment the train set is to use samples from a large data set that is weakly labeled or unlabeled [29, 116]. Also, we can aggregate samples from similar data sets that are larger or hallucinate samples using, for example, Generative Adversarial Networks [36]. The problem with data augmentation is that the rules are specific to each data set and can not be easily applied to other data sets.

Metric-Based

This technique represents data using a lower dimension (embedding), then uses simple models or distance functions to compare and classify the samples using their embedding [114].

This method can be classified into three types [114]. The first one is *multitasking learning* that learns several tasks at the same time using task-specific and task-generic information; the drawback is that every time a new task arrives, the model needs to be trained again [120, 128]. The second type is *embedding learning*, which embeds each sample in a smaller dimension so that in the embedded space, semantically similar samples are closer to each other. The embedding function is learned using the base set, and this method may not work when the FSL task is not related to the task used to train the function [101, 104, 110]. The last one aims to learn with *external memory*, which stores knowledge from the support set and then uses this to represent the test samples, also called query samples, in a weighted way. As external memory is limited, this method requires much external space and is computationally costly [75, 95].

Considering our final target, we found metric learning the most interesting few-shot approach; we believe that we can generalize better for CSAM. Also, those methods are non-parametric, which is desirable for the final few-shot task. Thus, in this Master's thesis, we explore embedding-learning methods to classify indoor scene imagery.

2.2.1 Aggregation Methods in Few-Shot Learning

In embedding-learning FSL approaches, the idea is to generate embeddings for each sample and then compare the query samples' embeddings with the support images' embeddings. The

comparison one by one is time-consuming: for each image in the query set, it is necessary to calculate the similarity between embeddings of all the images in the support set. To overcome this problem, Prototypical Networks [101] proposed to aggregate these embeddings and generate prototypes for each class in the support set. The prototype is the mean of the embeddings of all the samples in a class presented in the support set (Figure 2.2).

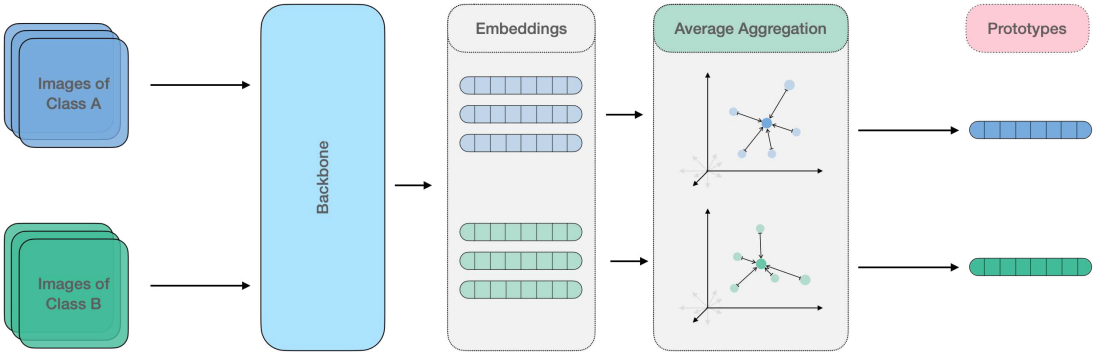


Figure 2.2: Representation of a prototype in few-shot learning.

Since then, many works have followed this idea and used prototypes to classify the query samples and the mean to aggregate the embeddings. However, as far as we know, no work investigates the impacts of the aggregation function on the prototypes. This can have a huge impact on the classification task.

Aggregating feature vectors in this way is seldom used in machine learning, apart from few-shot learning. However, this aggregation can also be seen as a pooling method, where we apply global pooling through all sample embeddings for each class and generate a final prototype. Thus, we explore pooling methods and aggregator functions to generate the prototypes.

Average [58] and max [11] pooling are the most used aggregation methods in the machine learning fields. Even though both methods showed good results in some benchmarks in the area, they have drawbacks. Average pooling (Equation 2.3) computes the mean of all elements in a region. This method does not present good results when the region contains multiple zeros; in an image, this would reduce the contrast of a region [126]. Max pooling (Equation 2.4) takes the max value of a region. This method shows problems when there are noises, and in images, sometimes the main element does not have the highest value; in those cases, the pooling would ignore completely the element [3].

$$f_{average} = \frac{1}{N}\sum_{i}^{N} x_i \qquad (2.3) \qquad f_{max} = \max\left(x_i, xi+1, ..., x_N\right) \qquad (2.4)$$

To overcome the problems of average and max pooling, other works propose a method that combines average and max pooling, trying to bring the best of those methods and suppress the problems. Mixed pooling [123] is defined by Equation 2.5, when $\lambda = 1$, it performs max pooling, and when $\lambda = 0$, the method performs average pooling. It showed superior results over average and max pooling, but as λ is a fixed value, sometimes it can lose important features [126]. Lp pooling [38] is also a method between max and average (Equation 2.6), when $p = \infty$, it performs max pooling, and when $p = 1$, it performs average pooling. This method can improve generalization, but its results depend greatly on the dataset.

$$f_{mixed} = \lambda f_{max} + (1 - \lambda) f_{average} \qquad (2.5)$$

$$f_{lp} = \left(\frac{1}{N} \sum_{i}^{N} x_i \right)^{\frac{1}{p}} \qquad (2.6)$$

Rank-based weighted pooling [100] is based on the probability of each value. It calculates the probability of each activation, and then, using these probabilities as weights, a weighted sum is performed over the activation values to take the aggregated value. Following the same idea, Sampaio Ferraz Ribeiro et al. [94] propose an aggregator method that uses self-attention to learn the better weights to aggregate the feature vectors, using a self-attention layer the attention values behave as a probability function that weights the activation values. The aggregator method is defined in Equation 2.7. Unlike the other works, which propose pooling layers, this work applies the aggregator the same way we are proposing.

$$f_{self-attn} = \sigma \left(\frac{Q(x)K(x)^T}{\sqrt{d_k}} \right) V. \qquad (2.7)$$

In FewTure [44], an FSL Transformer-based method used a LogSumExp function to aggregate vectors. Unlike our work, they do that for classification after comparing all the sample feature vectors. The function is defined in Equation 2.8.

$$f_{LogSumExp} = \log \sum_{i}^{N} \exp\left(x_i\right). \qquad (2.8)$$

In this Master's thesis, we explore the usage of five of these aggregator methods: average pooling, max pooling, Lp pooling, self-attention aggregator, and a LogSumExp function. We compare those aggregators for the indoor scene classification task.

2.2.2 Inductive and Transductive Learning

Inductive learning is the most common scenario in supervised classification. Its goal is to train a model using a labeled training dataset and then predict the labels of novel data. The idea is to learn a function to map any new example into the seen classes. In contrast, in the transductive setting, the model has access to the test set composed of unlabeled data and can extract statistical information about the data. Here, the goal is to predict only the classes seen in the test set [80]. In FSL, transductive learning uses the query set to extract and use information at training time. This approach has gained popularity since the publication of Liu et al. [66].

Recent FSL methods follow the transductive manner to deal with little data [7, 48, 49]. However, due to the sensitiveness of CSAM, it is desirable not to take parametrization into account at test time. Therefore, in this Master's thesis, we follow the inductive setting to learn how to predict the labels of novel data without seeing the test set.

2.3 Transformers

The Transformer model [108] was introduced in 2017 for machine translation tasks. It is an encoder-decoder architecture proposed to replace Recurrent Neural Networks (RNN) and Long

Short Term Memory (LSTM) [45] with a model based mostly on Attention modules. Until Transformers, RNNs, and LSTMs were state of the art for sequential tasks, these types of networks processed data sequentially. With that, two problems appear: difficulty correlating information that is distant in the sequence and inability to parallelize the training procedure. Using self-attention modules, Transformers solves the problems with computational complexity, parallelism, and long-range network dependencies. The usage of self-attention makes the network not depend on sequential processing anymore. Since it was proposed, Transformers and its variants have become state-of-the-art in several Natural Language Processing (NLP) tasks.

The core concept of Transformers architecture is the usage of attention layers. Attention was first introduced by Bahdanau et al. [6]. Their idea was to overcome the bottleneck of fixed-length vector encoding vector; for that, they proposed attention as a way for the model to search for relevant parts of the sentence to predict the target word. In their proposal, the attention is computed as follows:

$$\alpha_{ij} = \frac{\exp\left(f(s_{i-1}, h_j)\right)}{\sum_{k=0}^{K} \exp(f(s_{i-1}, h_k))}, \tag{2.9}$$

where s_{i-1} is the current decoder state, $h_0, ..., h_k$ are the encoder hidden states and f is the learnable function of the model. They use this attention to compute the context c that will serve as input to the next decoder step:

$$c_i = \sum_{k=0}^{K} \alpha_{ik} h_k. \tag{2.10}$$

After that, other attention designs were proposed [70, 103], but the usage of attention became more popular with the design proposed by Vaswani et al. [108]. Their design is called *scaled-dot-product attention*, or self-attention.

The idea of the attention is mainly to generate better embeddings for the words. In self-attention, the environment context is used to generate better embeddings, so based on the context, the attention mechanism uses the relation between tokens to give weights to the embeddings. This makes strongly related tokens closer (more similar), while tokens with a weak relation receive low weights, putting them far from each other (less similar).

The self-attention mechanism is defined in Equation 2.11. Here, Q, K, V are matrices called *Query*, *Key* and *Value*; they are generated by multiplying itself by the embedding input; and d_k is the *key* dimension. Together, *Query* and *Key* matrices generate a linear transformation of the embeddings, for that they are compared through dot product ($Q \cdot K^T$), resulting in the attention matrix, as the result can have high values, it is scaled dividing it by $\sqrt{d_k}$. Then, these result scores are multiplied by the vector V. This multiplication works as a weighted average of the values with different weights for each position. This mechanism is what "feels like" attention so that the model can focus on the right input [40]. Figure 2.3 shows a diagram of the attention process.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V. \tag{2.11}$$

In the Transformers architecture, the attention is implemented as a multi-head attention

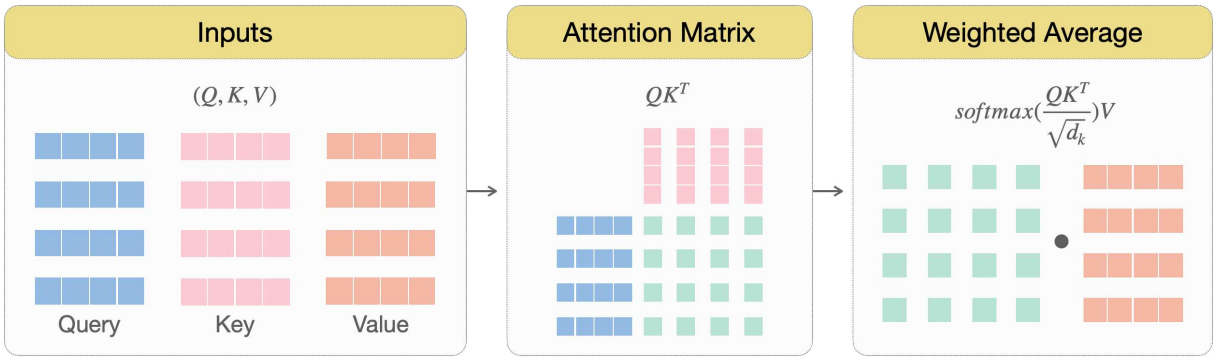


Figure 2.3: Attention mechanism architecture. First, Q, K, and V are taken as the input. Then, the dot product of Q and K generates the attention matrix. This matrix is then used to generate the weighted average of V. Figure from Ribeiro and Ponti [89].

layer, which is a layer composed of various self-attention layers working in parallel. This enables the model to consider attention in multiple subspaces, generating several transformed embeddings. The resulting attention matrix of each attention head is concatenated. Then, using a linear layer, the result is projected to the original dimension. This linear layer defines the weights to give to each attention matrix. This procedure helps the model to focus on several relevant positions for a given input.

Transformers were first proposed for machine translation, and as other methods [45] is composed of an encoder-decoder architecture. The encoder comprises a sequence of blocks with multi-head attention, normalization, and feed-forward layers. The idea of the encoder is to learn good embeddings to serve as input to the decoder. The decoder, on the other hand, is composed of two multi-head attention followed by normalization blocks, then it goes to a feed-forward layer; the difference is that in the decoder, the second multi-head layer receives the *Query* and *Key* from the output of the encoder, combining it with the output of the first multi-head. This procedure helps the model to generate the output based on the context learned from the encoder. Figure 2.4 shows the architecture of a Transformer.

Transformers have been widely explored in NLP. Usually, the models based on Transformers are pre-trained on extensive datasets. The Bidirectional Encoder Representations from Transformers (BERT) [25], and its variations [52, 67], use only the encoder architecture and pre-train the model using self-supervised learning. That way, the model does not need a large annotated dataset. Another successful work is the Generative Pre-trained Transformer (GPT) [87] that uses masked attention mechanisms and is based on Transformers' decoder architecture.

Due to Transformers' success in NLP, researchers have recently started exploring the usage of Transformers in Computer Vision. The first Transformers model proposed was Vision Transformer (ViT) [28]. This architecture is entirely based on the original Transformer encoder. However, instead of words, this model uses images as input. Since the original architecture was designed to use sequence as inputs (to work with words in sentences), to adapt the network to images, the authors proposed to split the images in patches of 16×16, Figure 2.5 shows this procedure. Each image patch is treated the same way as a token in the original Transformer.

The model showed promising results, surpassing the state-of-the-art when trained on a vast dataset (composed of 300M images). After that, several works were proposed to Computer Vision Transformers [14, 20, 106]. These techniques generally use the encoder of Transformers as

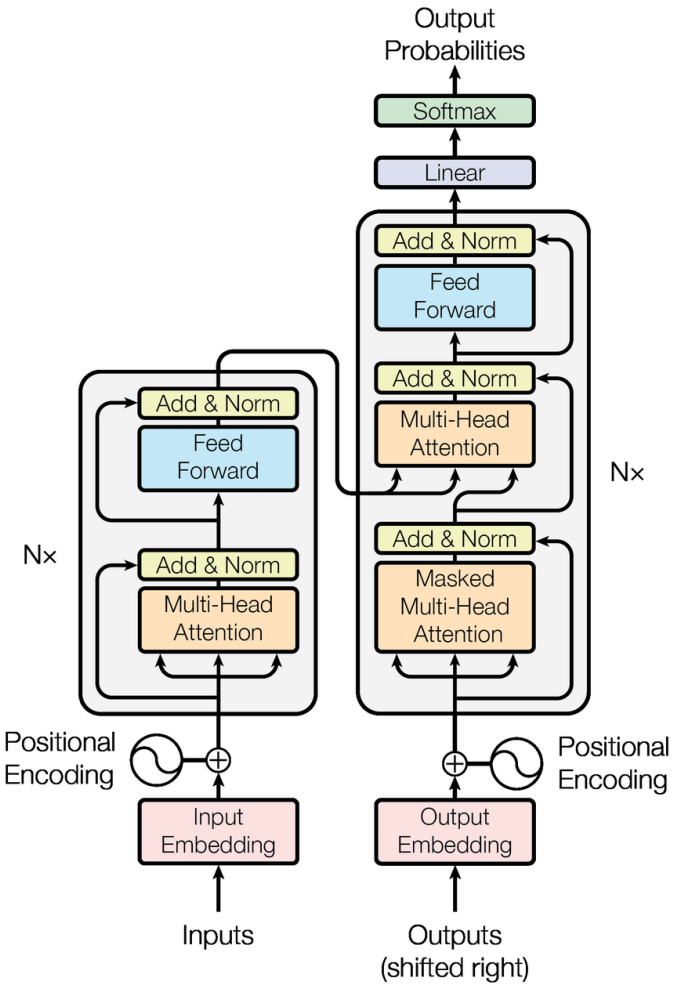


Figure 2.4: Transformers architecture. Figure from Vaswani et al. [108].

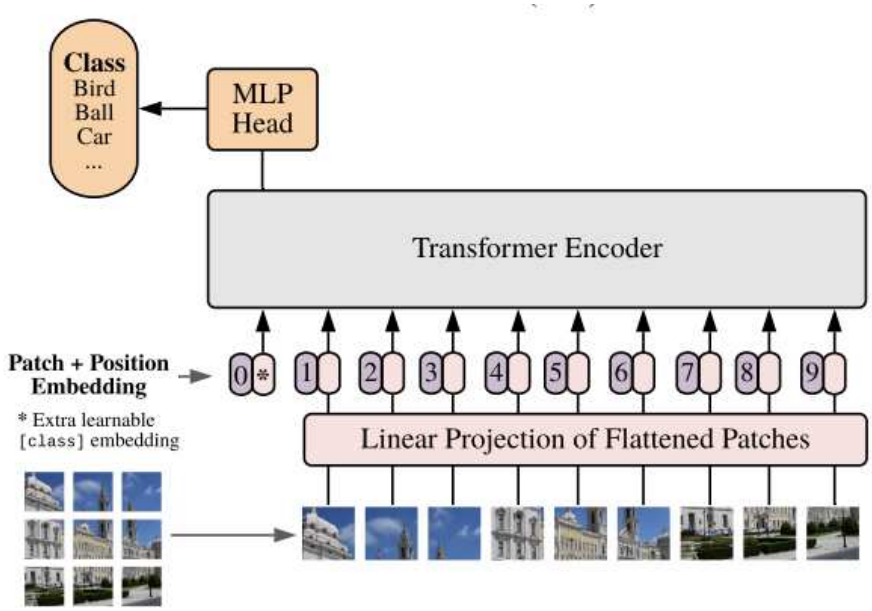


Figure 2.5: Vision Transformers image patch generation. Figure from Dosovitskiy et al. [28].

feature extractors. Furthermore, some works combine transformers with Convolutional Neural Networks (CNNs).

These works show that self-attention mechanisms can benefit Computer Vision tasks, mainly in fine-grained classification, because the model can focus only on parts of the image relevant to the classification. Unlike CNNs, attention mechanisms are not rigid and converge the model using fewer layers [21]. This process can benefit indoor scene classification, making the model focus on the relevant parts of the image, such as objects present in the scene.

Even though it is promising, research in attention to Deep Learning is new [21], and pre-training still requires a massive amount of data. The reason for that is the lack of inductive bias in Transformers, which differs from CNNs, where locality, 2-D local structure, and translation equivalence are present throughout the whole network. In ViT, we do not have this information since self-attention layers are global, so the spatial relations between patches need to be learned from scratch. This lack of inductive bias makes it very hard to train them with little data, but solutions from DeiT [106] to the few-shot ones we will discuss [27, 43, 47, 62] have found ways to work around this issue.

In this Master's thesis, we explore the usage of Transformers as backbone and self-attention modules as feature adapters to adapt the embeddings after extracting them using a CNN.

Chapter 3

Related Work

In this chapter, we overview the literature for scene classification (Section 3.1) and CSAM classification (Section 3.2). We also review the literature for embedding learning models (Section 3.3), a type of metric-based few-shot learning focusing on Transformers-based models.

3.1 Scene Classification

Scene classification, or scene recognition, is a computer vision task to classify indoor and outdoor environments. This task has been studied for many years and can be used in surveillance, autonomous driving, and robotics applications. Even though it is a well-studied task, recognizing indoor scenes remains a challenge [78]. In outdoor classification tasks, the model can focus on global spatial properties, while in indoor scenes, the model needs to exploit local and global properties [78]. Qiu et al. [85] show that some objects are essential to classify an indoor scene. For example, if we remove the bed from the bedroom, the scene will probably be classified as a living room or an office.

For Places365 dataset [129], InternImage [113] model reached 61.2% top-1 accuracy results (the state-of-the-art). The "FOS Fusion of Object and Scene Network" (FOSNet) [99], a CNN network, reached 90.3% on MIT Indoor [86] and 77.3% on SUN-397 dataset [118]. On SUN-397, the "Semantic-Aware Scene Recognition" [68] reached 74.0%; both works are supervised methods.

Regarding a Zero-Shot Learning (ZSL) scenario, one of the benchmarks was built on top of SUN-397, adding attributes to the images; this dataset is called SUN Attribute [79]. For that task, "Feature Transformation (T-Feature) Variational Autoencoder Generative Adversarial Networks" (TF-VAEGAN) [76] reached 66.0% top-1 accuracy. Another VAEGAN architecture feature-VAEGAN (f-VAEGAN) [117] achieved 64.7%; they claim to be a technique that can be applied to any-shot learning. More recently, "Selection using Proximal OpTimization" (SPOT) [37] proposed a model agnostic method that uses reinforcement learning to feature selection, that combined with VEAGAN, achieves 66.0% of accuracy. As these works were proposed to ZSL, they use semantic information from the dataset. Besides those works, as far as we know, no work has been proposed for scene classification using FSL, especially evaluating indoor scenes.

In "Leveraging Self-Supervised Learning for Scene Recognition in Child Sexual Abuse Im-

agery" [107], the author compared several self-supervised approaches for indoor scene classification, in particular, they proposed a dataset derived from Places [129] containing only indoor scenes that are prone to appear in the context of child sexual abuse imagery; The self-supervised approach reaches 71.6% balanced accuracy in the proposed dataset. In this work, we evaluate our method in the same dataset proposed by Valois et al. [107]. This study is the closest we have from the perspective of the final task.

From the perspective of few-shot learning, as far as we know, we are the first work to apply those methods to indoor scene classification. For that, we are using two subsets of Places365, one for indoor classes proposed by Valois et al. [107], and another with all the classes besides those presented in the previous dataset.

3.2 CSAM Classification

Most CSAM detectors are based on hashes [10, 23, 50, 81]. Those methods are great for identifying material reappearance but are sensitive to changes. To build more robust detectors, some works focused on machine learning; the proposed methods find different strategies to tackle CSAM detection since datasets for the task are not publicly available.

In this matter, Vitorino et al. [111] proposed the first deep neural network, where they used a pre-trained network and performed a 2-tiered transfer learning. In the first step, they transferred the learning for detecting pornography and then fine-tuned the CSAM detection model.

Other works proposed combining adult detection with age estimation. Macedo et al. [71] use Yahoo's open source pornography detector [72] and combine this network with a network trained with faces to determine age group, gender, and if the face is from an adult or not. Following the same idea, some works also used neural networks to estimate age in combination with adult content detection [4, 16, 34, 90, 91]. Chaves et al. [16] improved age estimation by combining the input of original faces with occluded eyes; they argue that this training approach improves the estimator's performance. Moreover, to improve the performance of both classification tasks, Gangwar et al. [34] proposed a network with an attention mechanism. Besides classifying images, Rondeau et al. [91] combines the model with a frame extractor; that way, the network can automatically extract frames from a video and classify them for CSAM. In addition to combining adult content classification with child detector, Dalins et al. [22] implement an additional method to determine CSAM level in ten categories, from *no sexual activity* to different *levels of child abuse*, passing through *adult content*.

Focusing on videos, Westlake et al. [115] and Brewer et al. [12] use biometric features, face, and voice, using them to link with other videos having the same faces or voices; this could be used to find media associated with an investigation, identifying victims and offenders.

Even though most works focus on age estimation in combination with adult content detection, CSAM is complex enough to require combining more approaches for automatic detection [59]. To bring insights on what could be done to help CSAM detection, Laranjeira da Silva et al. [57] proposed an analysis template to understand CSAM images without seeing them. In their analysis, they show the distribution of some features such as age, gender, face detection, ethnicity, scenes, and objects; the analysis was demonstrated on Region-based annotated Child Pornography Dataset (RCPD) [71], which is a Brazilian CSAM benchmark proposed in collab-

oration with the Federal Police. The analysis showed that context information from objects and scenes is correlated with CSAM.

Unlike most of those works, in this Master's thesis, we do not aim to classify CSAM directly. Our objective is to help CSAM investigation with the triage of possible material candidates to be analyzed. To do that, we classify indoor scenes, the environments where those materials are mostly recorded.

3.3 Embedding Learning

As mentioned in Section 2, this Master's thesis focuses on metric-based few-shot learning. More specifically, we concentrate on embedding learning models. Embedding learning approaches aim to learn an embedding function so that the embedding for each sample is closer if the samples are similar.

First, we need to learn an embedding function from the base set to classify the samples with embedding learning models. Then, we use the learned embedding function in test time to generate feature embedding for the support and query set. After embedding each sample, we compare the query and support samples using a similarity function. If the samples are close, they are from the same class. Figure 3.1 illustrates this process. Some works may use different embedding functions for the support and query set.

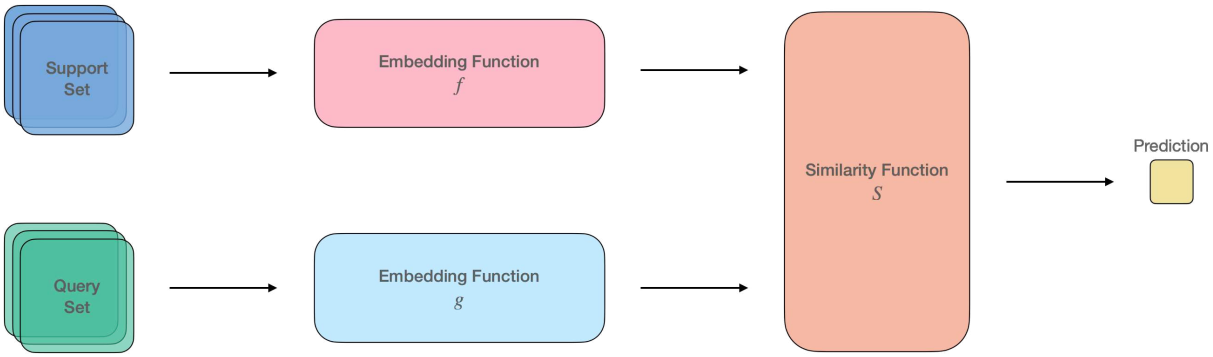


Figure 3.1: Embedding learning process. Figure adapted from Wang et al. [114].

In Table 3.1, we summarize the most popular embedding learning methods, highlighting with (*) the methods reproduced in this Master's thesis. We categorize them as task-specific, task-invariant, or hybrid, as in Wang et al. [114]:

- Task-specific models learn a specialized embedding function for each task. For that, these models use only samples from that task.
- Task-invariant models learn a general embedding function from the base set and then use this function on the few-shot task without retraining.
- Hybrid models adapt generic models, retraining the model using specific information from the few-shot set.

Matching Networks [110] use two embedding functions g and f, one to embed support samples and the other to embed query samples. Besides that, they introduce bidirectional

Table 3.1: Embedding learning methods. Methods with (*) are the ones we could reproduce.

	Category	Method	Dataset	Network Type	Backbone	Similarity Mesure	Conference	Code Available
CNNs-based	Task-invariant	Marching Networks [110]	Omniglot, ImageNet	CNN, LSTM	–	Cosine Similarity	NeurIPS 2016	No
	Task-invariant	ProtoNet [101]*	Omniglot, *mini*ImageNet, CUB	CNN	Conv-4	Euclidean Distance	NeurIPS 2017	Yes
	Task-invariant	Relation Network [104]*	Omniglot, *mini*ImageNet, CUB, AWA	CNN	Conv-4	Learned Distance	CVPR 2018	Yes
	Hybrid	TADAM [77]	*mini*ImageNet, FC100	Adaptative CNN	ResNet-12	Euclidean Distance	NeurIPS 2018	No
	Task-invariant	GNN [96]	Omniglot, *mini*ImageNet	CNN, GNN	Conv-4	Learned Distance	ICLR 2018	Yes
	Task-invariant	SNAIL [73]	Omniglot, *mini*ImageNet	CNN with Attention	–	Learned Distance	ICLR 2018	No
	Task-invariant	Baseline++ [19]*	*mini*ImageNet, CUB	CNN	ResNet18	Cosine Similarity	ICLR 2019	Yes
Transformers-based	Hybrid	FEAT [122]*	*mini*ImageNet, *tiered*ImageNet, OfficeHome	CNN and Self-Attention	ResNet-18	Cosine Similarity	CVPR 2020	Yes
	Hybrid	CrossTransformers [26]*	Meta-Dataset	CNN and Self-Attention	ResNet-34	Euclidean Distance	NeurIPS 2020	Yes
	Hybrid	URT [65]	Meta-Dataset, MNIST, CIFAR-10, CIFAR-100	CNN and Self-Attention	ResNet-18	Cosine Similarity	ICLR 2021	Yes
	Task-invariant	SSFormers [17]*	*mini*ImageNet, *tiered*ImageNet, CIFAR-FS, FC100	CNN and Self-Attention	ResNet-12	Custom	SCIENCE CHINA Information Science	Yes
	Hybrid	P>M>F (ProtoNet) [47]*	*mini*ImageNet, CIFAR-FS, Meta-Dataset	Transformers	ViT small	Cosine Similarity	CVPR 2022	Yes
	Hybrid	FewTure [44]	miniImageNet, tieredImageNet, CIFAR-FS, FC100	Transformers	ViT Small	Cosine Similarity	NeurIPS 2022	Yes
	Task-invariant	HCTransformers [43]	miniImageNet, tieredImageNet, CIFAR-FS, FC100	Transformers	ViT small	Linear classifier	CVPR 2022	Yes
	Hybrid	SUN [27]	miniImageNet, tieredImageNet, CIFAR-FS	Transformers	Visformer	Cosine Similarity	ECCV 2022	Yes
	Hybrid	SP [18]	miniImageNet, tieredImageNet, CIFAR-FS, FC100	Transformers	Visformer	Cosine Similarity	CVPR 2023	Yes
	Task-invariant	SMKD [62]	miniImageNet, tieredImageNet, CIFAR-FS, FC100	Transformers	ViT small	Linear classifier	CVPR 2023	Yes

LSTMs [45] (Long Short Term Memory networks) to embed each image depending on the embedding of the others. The Prototypical Networks (ProtoNet) [101] introduce the idea of using prototypes to represent the feature embedding of each class. These prototypes are the average of all samples embedding in that class. In addition, this work's embedding functions for the support and query set are the same. ProtoNet is not a new network; it proposes a way to classify the samples faster without comparing all samples. This pipeline can be used with different backbones; the state-of-the-art method (P>M>F [47], a pre-training > meta-training > fine-tuning pipeline) uses ProtoNet with a different backbone. The Relation Network [104] also uses prototypes to represent each class. However, to classify the samples, they introduce the relation module, a convolutional neural network, and its output is a relation score in the interval $[0, 1]$, where one indicates that the sample belongs to that class.

"TAsk-Dependent Adaptive Metric" (TADAM) [77] is also based on prototypical networks, but they add two complexity layers. The first layer is a pre-softmax scaling, which learns the temperature coefficient to adapt softmax. The second is task conditioning, a network that learns to choose the best parameters for the feature extractor from each task. In "Few-Shot Learning with Graph Neural Network" (GNN) [96], the authors followed the idea of ProtoNet generating prototypes for classes and to classify the samples, they proposed the usage of a GNN. The "Simple Neural AttentIve Learner" (SNAIL) [73] combines temporal convolution and soft-attention layers to overcome the problem of building a very specialized network. To do so, the temporal convolution aggregates information from the base set, and the soft attention identifies specific information from samples. The same network generates the embedding function and classifies the samples.

In "A Closer Look at Few-Shot Classification" [19], the authors compare several works and conclude that using more robust backbones reduces the performance gap between methods when the base and support set domains are similar. Otherwise, they propose Baseline and Baseline++, exploring a trainable classifier that continues to learn during the meta-testing stage. Baseline uses a linear layer as a classifier, while Baseline++ uses a cosine distance.

Recent research focuses on task-invariant and hybrid embedding models because it is more interesting to use larger datasets to train the model and then apply it to the few-shot task. Besides, to train a task-specific model, more data from the task is required [114], and CSAM is constrained on both the number of samples and access. For these reasons, we benefit from not fitting parameters during the test. Therefore, this Master's thesis explores both task-invariant and task-hybrid. As most Transformers-based approaches are hybrid, we believe this is the best model type for our problem.

Although this work focuses on metric-based few-shot learning, some optimization-based models are relevant to the literature and showed good results. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks" (MAML) [32] proposes an algorithm that is task-agnostic. Using a small amount of data, this algorithm learns a good parameter initialization for a new task with only a few gradient steps. Rusu et al. [93] argue that it is difficult to have a generalized model using only a few samples to compute the gradients, given that the gradients are in a high-dimensional space. To solve this issue, they propose Latent Embedding Optimization (LEO), which learns low-dimensional latent embedding parameters, and then the model performs the optimization in this space.

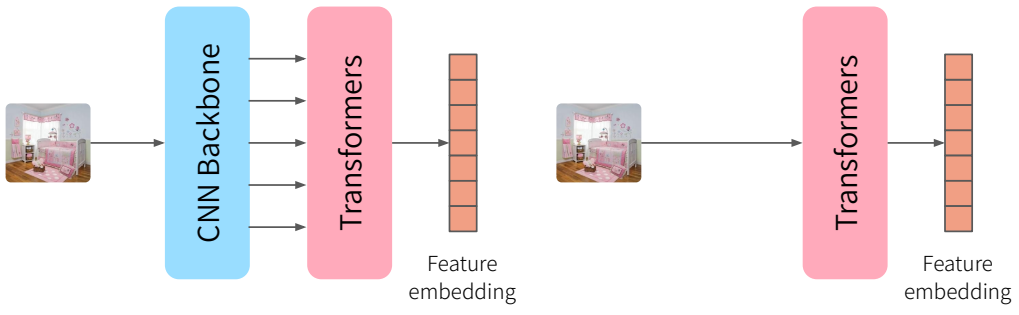
All the metric-based works are relevant for indoor classification, but we select only a few

to reproduce. As Matching Networks [110], TADAM [77], and SNAIL [73] do not have their codes available, we did not reproduce them. One of our objectives is to compare purely convolutional with Transformer-based networks; for that reason, graph neural networks [96] are not the focus of this Master's thesis. Finally, we were able to reproduce ProtoNet [101], Relation Network [104], and Baseline++ [19] as CNNs-based few-shot learning. Those works are important for understanding how purely convolutional networks perform for indoor scene tasks. Even though optimization-based are parametric, which is not desirable for CSAM, we also reproduce them for indoor scene classification, as they are relevant to the few-shot literature.

Transformers in Few-Shot Learning

In 2017, Transformers were first proposed for machine translation [108]. However, only in 2020 did the method begin to be applied to Computer Vision [28] and rapidly achieve state-of-art results for several tasks. Based on the good results achieved by the architecture, some works started to study the use of transformers in FSL vision tasks.

Transformers in FSL started being used to adapt the features that were extracted using a CNN [17, 26, 65, 122]. More recently, Transformers started being explored as a feature extractor [18, 27, 43, 44, 47, 62], which is a challenge, since Transformers tend to overfit when trained with a small dataset. The difference between these two approaches is that when using Transformers as a feature adapter, we first extract the features from a purely convolutional neural network and then adapt those features using a Transformers-based network; in most cases, self-attention mechanisms; this way, the model can be trained with fewer data and the CNN bring the inductive bias needed; The other way is to use Transformers directly to extract the features, without the need of a CNN. As Transformers lack inductive bias, the training process needs to be adapted, or the pre-training stage needs to be done using a large dataset. Figure 3.2 exemplifies both approaches.



(a) Feature adapter (b) Feature extractor

Figure 3.2: Usage of Transformers in Few-Shot Learning models.

In Few-shot Embedding Adaptation with Transformer (FEAT) [122], the authors argue that the embeddings of task-specific few-shot methods did not generate a good representation for the target task based only on the learning representation from the base set. Thus, they were the first to propose a set-to-set function to adapt the feature embedding for the few-shot task. The idea is to separate the embedding spaces so that the visual features are most discriminative for a given task. For that, they compared the usage of Transformers with Bidirectional LSTMs [45],

DeepSets [127] and Graph Convolutional Networks [54, 96]; Transformers-based adaptation showed better results. CSAM classification could benefit from the set-to-set function since the model is trained on an indoor scene dataset and needs to adapt the learned features for the indoor scenes presented in CSAM.

The authors of CrossTransformers [26] argue that CNNs used as a backbone for ProtoNet represent only the base set, losing information that could help transfer learning to the target task; they called this phenomenon supervision collapse. To overcome this problem, they proposed two methods. First, they used self-supervised learning to maintain intra-class variation and then better transfer the knowledge to new tasks. Second, they proposed a Transformer-based architecture to help the classification using local features that may be similar to the features from the target task. We can benefit from using local features, as the Trasformer-based architecture can help us focus on objects present in the scene, and the self-supervised model proposed can be great for transferring the learned knowledge to the CSAM environment.

The "Universal Representation Transformers" (URT) [65] proposes using a universal representation layer, which extracts features from several backbones pre-trained on different datasets. They combine these embeddings using a multi-head self-attention layer to integrate representations from multiple domains and adapt it to generate a representative embedding from the few-shot samples. Although using several backbones can be time-consuming, the embedding representation from backbones pre-trained with different datasets can generate a great representation for our task.

In the "Sparse Spatial Transformers for Few-Shot Learning" [17], the authors argue that most metric-learning works use global features, which lose local information in the representation, or dense feature representations, that lose contextual information. To solve that, they propose Sparse Spatial Transformers (SSFormers). This network generates features that maintain local and contextual information and can find relevant features while suppressing irrelevant features. They also propose a novel similarity function called patch matching. For indoor scene classification, as mentioned before, it is important to keep local and contextual information; for this reason, we can benefit from SSFormers for our task.

Hu et al. [47] starting from a ProtoNet pipeline, investigate three design choices: pre-training data, backbone architecture, and fine-tuning during meta-testing. They conclude that pre-training with a large dataset, such as ImageNet-1K [24], improves the results compared to the conventional pipeline with no pre-training; we arrive at a similar conclusion for our task. Also, Transformers outperforms other backbones when pre-trained in such large datasets. Lastly, they observed that fine-tuning on the support set is more helpful in out-of-domain scenarios. Based on those conclusions, they proposed a pre-training > meta-training > fine-tuning pipeline, which they named P>M>F. This pipeline is the same as ProtoNet, using a ViT as the backbone. This is the first few-shot learning work to use ImageNet-1K [24] that we can generate more robust embeddings for indoor scene classification, and using a more robust pre-trained backbone can also generate better embeddings for the CSAM. This work's fine-tuning stage is irrelevant since we do not want to use extra information during meta-testing.

Except for SSFormers [17], which calculates similarity from all sparse attention patches, the other works presented until now generate a prototype to classify the samples. That makes the model less computationally costly, allowing it to run in constrained environments like the ones available for CSAM.

The authors of "Rethinking Generalization in Few-Shot Classification" [44] proposed a few-shot classification with Transformers Using Reweighted Embedding (FewTURE). A model that uses as backbone a pre-trained self-supervised ViT, that learns how to weight the importance of the tokens for a given task. They fine-tune the model during meta-testing to adapt the important weights to the given task. They compute the cosine similarity between all the support tokens and the query tokens; then, they correlate these similarities with the token importance weights to determine which tokens are more relevant for the classification.

In Attribute Surrogates Learning and Spectral Tokens Pooling in Transformers for Few-shot Learning [43], they propose HCTransformers, a hierarchically cascaded Transformers architecture, this network is trained using a base set in a self-supervised manner. They also propose a spectral token pooling that extracts intrinsic image structures and optimizes parameters via attribute surrogates. Semantic Prompt for Few-Shot Image Recognition [18] uses extra semantic information from the classes' names to combine semantic and visual information to find token regions' class-specific features.

In "Self-Promoted Supervision for Few-Shot Transformer" (SUN) [27], they propose a framework to train ViTs in a few-shot learning environment. The idea is that ViT needs lots of training data to overcome the problem with a lack of inductive bias, but with their framework, we can pre-train the ViT on the few-shot learning dataset. To that, they propose a meta-training stage where they employ global supervision and location-specific supervision to generate token pseudo labels; this tells the ViT which tokens are similar; this step is learned from the whole base set. Then, they perform meta-finetuning, using the same weights learned in the pre-training stage; they fine-tune the model using few-shot tasks, that is, n-way k-shot support samples, to classify query samples. This classification is done the same way as in ProtoNet.

The Supervised Masked Knowledge Distillation for Few-Shot Transformers (SMKD) [62] proposes the use of self-supervised learning combined with supervised self-distillation to be able to train a ViT using small datasets.

The networks proposed by FewTURE [44], HCTransformers [43], SUN [104], and SMDK [62] are heavy, needing robust GPUs, for this reason, we could not reproduce those works. Even though these methods could achieve good results for indoor classification, running these methods in a constrained environment, such as the environment for CSAM testing, would be unfeasible.

In this Master's thesis, we explore the usage of the Transformer architecture in feature embedding models for indoor scene classification and compare it with purely convolutional approaches. To the best of our knowledge, no prior work in the field of few-shot learning is dedicated to indoor scene classification.

Chapter 4

Datasets

This chapter describes the datasets used in this Master's thesis. In Section 4.1, we detail the datasets to pre-train the backbones. In Section 4.2, we present the base dataset for fine-tuning or training the network for the few-shot task. Then, in Section 4.3, we describe the datasets used for evaluation. In Section 4.3.1, we present the dataset used for evaluating the network in the indoor classification task. In Section 4.3.2, we describe the dataset used for the OOD Scenes test, which evaluates the network's ability in uncontrolled scenarios. In Section 4.3.3, we present the CSAM dataset for indoor scenes' classification, used to analyze the proposed model on the CSAM task. Last, in Section 4.3.4, we present a benchmark for CSAM classification. Table 4.1 summarizes all the datasets used in each step.

Table 4.1: Summarization of the datasets used for pre-training, meta-training, and evaluation. Usage stands for the step we used in the dataset, and size is the total number of samples.

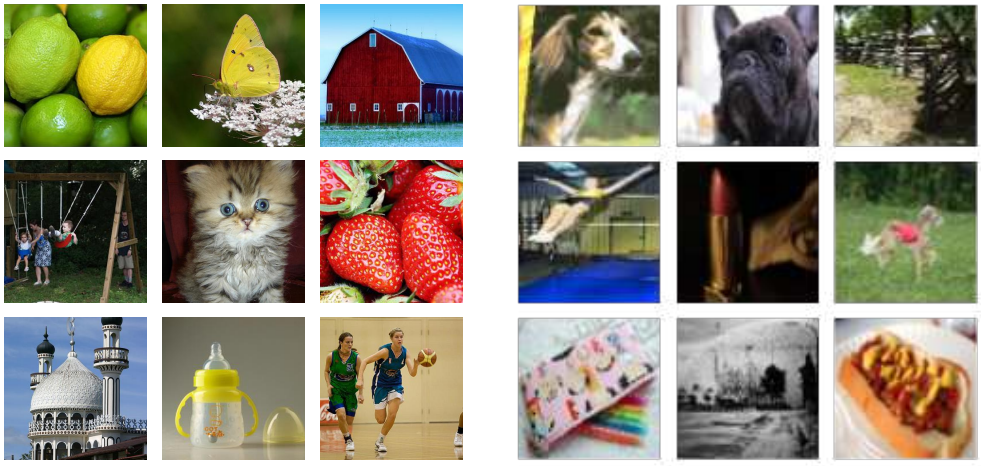
Usage	Dataset	Size	# Classes	Year
Pre-training	ImageNet-1K [24]	1,200,000	1000	2012
	MiniImageNet [110]	60,000	600	2016
Base Dataset	Places600 (ours)	160,800	268	2023
Evaluation	Places8 [107]	407,640	8	2022
	OOD Scenes [107]	80	8	2022
	CSAM (Brazilian Federal Police)	46,006	6	2022
	RCPD [71]	2,138	2	2018

4.1 Pre-training Datasets

ImageNet-1K [24] was proposed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. It contains 1000 object classes from natural images. The training dataset consists of more than 1.2M of images. This dataset was proposed for object classification and localization. It is used in different machine-learning tasks. Therefore, it is easy to find models pre-trained on this dataset. Figure 4.1a shows some samples.

***Mini*ImageNet [110]:** is a subset of ImageNet-1K. It contains 100 classes randomly selected

from the ImageNet ILSVRC 2012, each class containing 600 images of size 84×84 pixels. The classes are split into 64 base classes, 16 validation classes, and 20 novel classes. The dataset is a benchmark for few-shot learning. Figure 4.1b shows some samples.



(a) ImageNet-1K (b) *Mini*ImageNet

Figure 4.1: Samples of the base datasets: ImageNet-1K and *mini*ImageNet.

4.2 Base Dataset

The base dataset is used to train or fine-tune the model in the few-shot pipeline. Unfortunately, there is no few-shot benchmark for scene classification, so we used Places365-Challenge dataset [129] as our base dataset.

Different from standard machine-learning pipelines, in the few-shot the classes of the base set need to be representative but disjointed from the classes used for evaluation (also named support and query sets). For that reason, we could not use Places365 entirely since some of the indoor classes presented in this dataset are used for evaluation.

We used a subset of the Places365 training split to deal with that. In that subset, we removed all the classes presented in the evaluation dataset, explained in Section 4.3.1.

Beyond this restriction, we also want a comparable few-shot dataset, so we selected only 600 samples per class of the provided subset to have a dataset similar to *mini*ImageNet in size. Some of the classes of Places have less than 600 samples, so, in those cases, we decided not to include this class in the resulting subsample. After filtering all samples, we ended up with a dataset that contains 268 classes and 600 samples per class; we named this dataset **Places600**. In Appendix A, there is a list containing all the classes' names. Figure 4.2 shows some samples.

Figure 4.2: Samples of Places600 dataset.

4.3 Evaluation Datasets

4.3.1 Places8

For evaluation, we used Places8 proposed by Valois et al. [107]. In this work, with the assistance of police agents, they selected 23 indoor scenes from Places365-Challenge [129] and grouped those scenes into 8 classes. These classes are: *bathroom*, *bedroom*, *child's room*, *classroom*, *dressing room*, *living room*, *television studio*, and *swimming pool*. Table 4.2 presents the number of samples in each class and the original categories from Places365, and Figure 4.3 shows some samples.

Table 4.2: Description of Places8 dataset. The table presents the classes and the number of samples in each dataset split: test, training, and validation; also, it shows the original categories of the samples from Places365. Table reproduced from Valois et al. [107].

Class	Test	Training	Validation	%	Original Categories
bathroom	5,740	51,655	200	13.4	bathroom, shower
bedroom	11,112	100,012	600	25.9	bedchamber, bedroom, hotel room, berth, dorm room, youth hostel
child's room	4,650	41,849	300	10.8	child's room, nursery, playroom
classroom	3,751	33,763	200	8.7	classroom, kindergarten classroom
dressing room	2,432	21,889	200	5.7	closet, dressing room
living room	9,940	89,458	500	28.7	home theater, living room, recreation room, television room, waiting room
studio	1,404	12,633	100	3.3	television studio
swimming pool	1,505	13,547	200	3.5	jacuzzi, swimming pool
Total	40,534	364,806	2300	100	

This dataset is used for validation and test. We wanted more samples than were available by the validation split for validation. For that reason, we used the train split for validation. We randomly sampled 600 samples from each class of the train split. The number of samples was

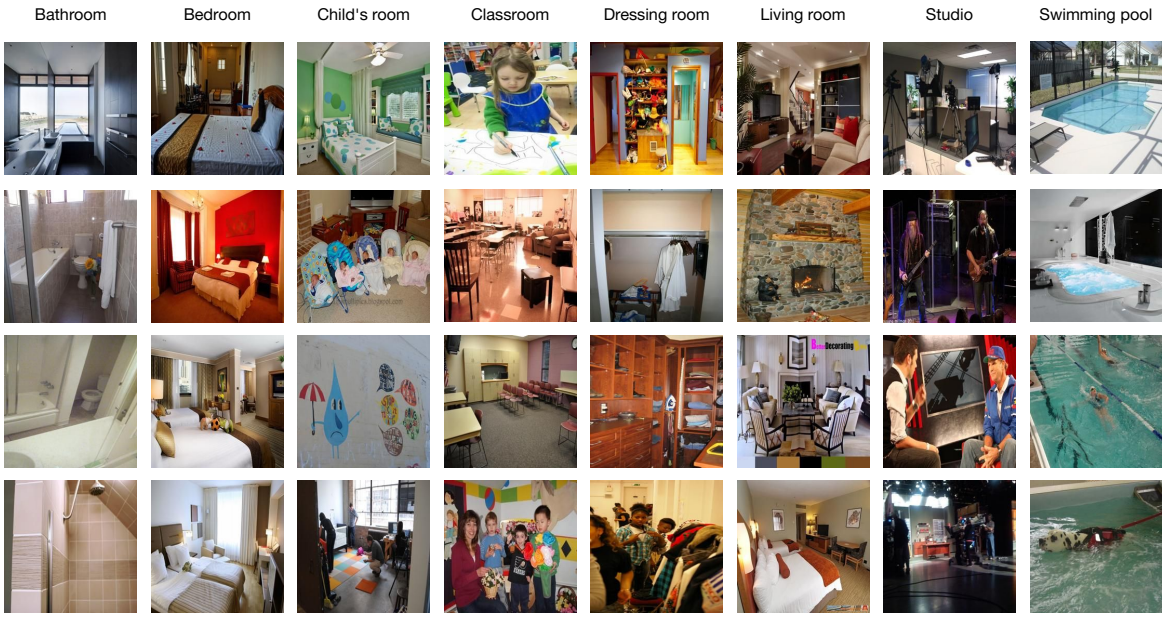


Figure 4.3: Samples of Places8 dataset.

selected based on *mini*ImageNet. For testing purposes, the split used was the same test split proposed by Valois et al. [107].

4.3.2 OOD Scenes

To understand the generalization ability of the chosen model, we evaluate our model in OOD Scenes dataset. The purpose of this evaluation is to use images from the same classes, but that came from a non-controlled environment, as in Places8.

We use the same dataset proposed by Valois et al. [107] for the evaluation. That way, we can compare the generalization of the model. The dataset comprises 10 images per class, with the eight classes from Places8: *bathroom*, *bedroom*, *child's room*, *classroom*, *dressing room*, *living room*, *television studio*, and *swimming pool*. The images were collected from Google Images, Bing Images, and the Dollar Street dataset.

Even though this is a small dataset, their images represent different scenarios than Places8. Figure 4.4 shows some samples.

4.3.3 CSAM Dataset

To evaluate our model in a CSAM environment, in collaboration with the Brazilian Federal Police, a forensic expert created a scene-labeled dataset from CSAM Imagery. This dataset is only accessed by agents from the Federal Police, so we only used this dataset as a final evaluation of the best model. This dataset is under construction and has not yet been published, so it is not available for testing outside the research group.

Originally, this dataset contained 46,006 samples, 45,970 samples for training, and 4,592 for testing. Those samples are classified into *CSAM*, *suspected CSAM*, *drawing*, *people*, *porn*, and *other*. From the test set, 615 samples were randomly selected from manual labeling across

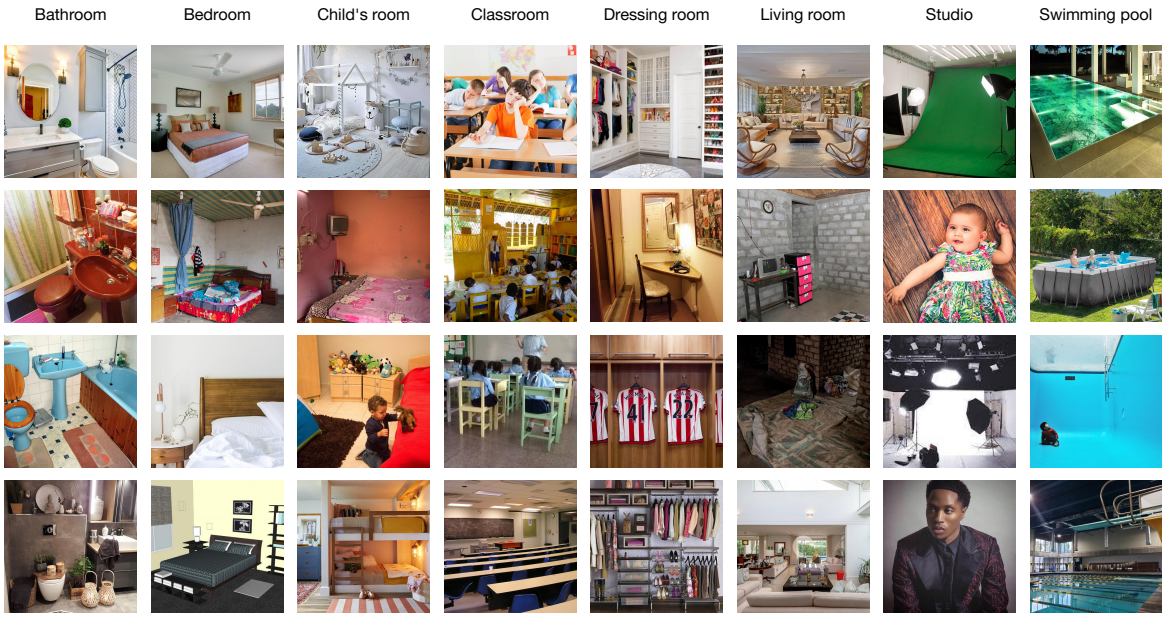


Figure 4.4: Samples of OOD Scenes dataset.

the eight classes from Places8 (Section 4.3.1. From those, only 374 samples did contain indoor scenes; unfortunately, the classes found were from six of the eight indoor classes: bathroom, bedroom, child's room, living room, studio, and swimming pool.

4.3.4 Region-based Annotated Child Pornography

To understand the importance of the features of indoor scenes to the CSAM classification, we evaluate our method on Region-based Annotated Child Pornography (RCPD) [71], a dataset focused on the direct CSAM classification from the Brazilian Federal Police. This is not our main task, as we aim to classify indoor scenes, but we want to understand the behavior of the proposed model on this task. This dataset is also only accessed through agents from the Federal Police.

Computer forensic experts labeled the samples in this dataset. The images can have multiple labels, such as gender, age, and level of nudity. The binary classification of CSAM is based on the combination of those labels. The dataset contains 2138 samples, 837 are CSAM, and 1301 are not CSAM. This dataset allows researchers to submit their algorithms through a website[1].

[1] `http://www.patreo.dcc.ufmg.br/rcpd`

Chapter 5

Methodology

In this chapter, we explain our proposed methodology. The following sections describe the training and evaluation pipelines focusing on research reproducibility, which enables future extensions. In Section 5.1, we define the problem we are tackling, focusing on FSL embedding learning. In Section 5.2, we present the experiment design in the leans of the research questions. In Section 5.3, we describe the methodology followed in the evaluation stage of the model. Finally, in Section 5.4, we present the metrics used in this work. Our main goal is to build a few-shot learning model trained on indoor scene images to help classify CSAM.

5.1 Problem Definition

This Master's thesis is focused on using FSL for indoor scene classification. In FSL, we aim to classify unseen samples using only a few labeled data, which is called N-way K-shot classification, where N is the number of classes to be classified and K is the number of labeled samples per class available for the classification. For that, FSL methods generalize the knowledge learned from the base set D_{train} to unseen test data D_{test}, where the classes $C_{train} \cap C_{test} = \emptyset$. We employed the episodic meta-learning protocol for training and testing strategies, where episodes do training and testing. One episode E is composed of the support set $S_{sup} = \{S_{sup}^{nk} = \{x_{sup}^{nk}, y_{sup}^{nk}\} \mid n = 1, ..., N; k = 1, ..., K\}$ and the query set $S_{qry} = \{S_{qry}^{i} = \{x_{qry}^{n}, y_{qry}^{n}\} \mid n = 1, ..., N\}$. The idea of episodic training is that each episode mimics one few-shot task entirely; each episode is then composed of support (training set within an episode) and query (test set within an episode) set. First, we train the model through episodes, where each episode represents a task; for that, D_{train} is randomly sampled into E; this training process is called *meta-train*. During the test, our goal is to classify several tasks (episodes) from the test set to generate those episodes randomly sampled D_{test} into E; this testing process is called *meta-testing*.

More specifically, we follow the embedding learning approach (see Section 3.3) because those approaches employ non-parametric learning, which is very desirable to avoid fitting parameters to sensitive data. The idea of those methods is to embed each sample $x_i \in S \subseteq R^d$ in a new space dimension $z_i \in Z \subseteq R^e$, where d and e are different dimensions, in a way that in the new dimension Z similar samples are closer, while dissimilar samples are distant. To do that, those methods are composed of three components: a function g, which embeds the support

samples $x_{sup} \in S$ to a smaller dimension $z_{sup} \in Z$, a function f, which embeds the query samples $x_{qry} \in S$ to a smaller dimension $z_{qry} \in Z$, and a similarity function s, that measure the distance $g(x_{sup})$ and $f(x_{qry})$. Then, the sample x^n_{qry} is classified as the same class of x_{sup} that resulted in the minimum distance [114] (see Figure 3.1). As said in Section 3.3, the function f and g can be the same function.

To solve indoor scene classification with few-shot learning, we first compared existing embedding-learning few-shot methods applied to our task. Then, with the best model in hands, we thoroughly studied the method's hyperparameters and tested different backbone options aligned with our research questions (see Section 1.4) and final indoor scene classification task, which will be expanded in Section 5.2.

5.2 Experimental Design

Figure 5.1 illustrates the pipeline followed in our experiments. As Section 5.1 mentioned, reproducing existing FSL approaches was a big part of our efforts. One of our objectives (O2) is to compare two different few-shot learning approaches for indoor scene classification; these approaches are purely convolutional networks and models based on Transformers. We followed the training methodology proposed by each work, but their pipelines also fit the one presented in this section. They differ mainly on the backbone used, training protocol, and how they compare the features. For a comprehensive comparison, we also reproduced two optimization-based methods (see Section 3.3 for details on those methods): "Latent Embedding Optimization" (LEO) [93], and "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks" (MAML) [32], but those methods performed poorly compared to the embedding-learning models.

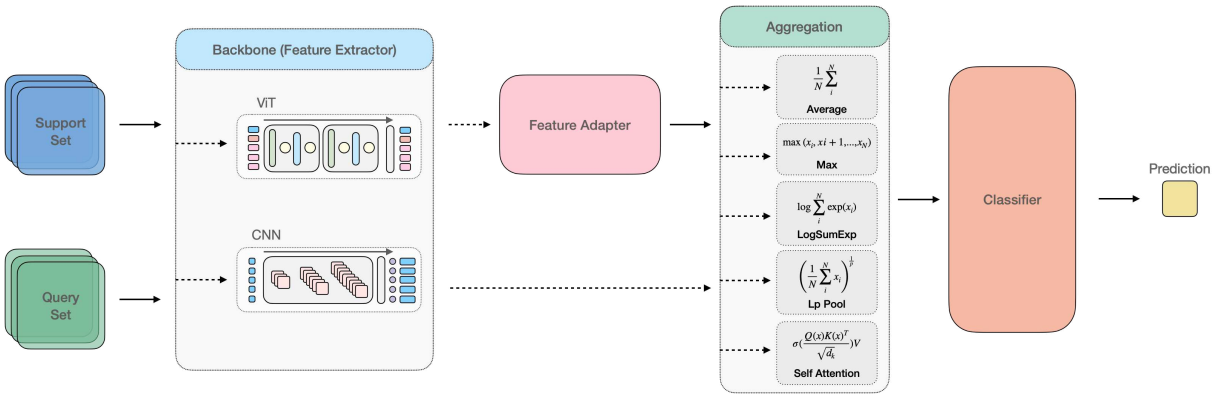


Figure 5.1: Experimental pipeline followed in this Master's thesis. The dotted line arrows represent the possibilities for the experiments, and the solid line arrows represent fixed steps in the experiments.

For all the methods reproduced, we only considered pre-trained models; most of the networks were pre-trained on *mini*ImageNet, except for P>M>F [47] (a pre-training > meta-training > fine-tuning pipeline), which was pre-trained on ImageNet-1K [24]. We also compared the pre-trained models with those fine-tuned on Places600 (Section 4.2).

This first experiment across the methods gave us answers for research question Q1, comparing purely convolutional methods with Transformers-based ones. Then, comparing the models

using Transformers-based approaches, they use Transformers in two ways: as the backbone of the model to extract the features [47], or to adapt feature extracted from a purely convolutional network, as ResNet [42] to a new task [17, 26, 122]. So, comparing those methods, we can answer the research question Q2.

After comparing the indoor scene classification methods, we focused on improving the results obtained following P>M>F [47], which gave us the best result. From this model, we defined our methodology from the lenses of our research questions.

Q1. **Do Transformers-based methods outperform convolutional neural network methods for indoor classification?** In addition to comparing the methods described above, we further explore this research question for P>M>F. For that, we need to experiment with changes in the backbone, comparing feature extractors that are purely convolutional with Transformers-based networks to evaluate what type of network can generalize better for the target task. For that, we considered two backbones: ResNet50 [42], which is a purely convolutional network, and Vision Transformers [28], both pre-trained on the same dataset. By comparing the previous methods, we can answer this question.

Q3. **What are the impacts of feature vector aggregators on few-shot models for indoor classification?** Except for MAML [32] and LEO [93], which did not lead to great results, all the embedding learning methods we are interested in are derivations of ProtoNet [101]. So, all these methods depend on aggregating the feature vectors extracted by class and distance computation of the features from the query set and those prototypes. Following ProtoNet [101], all these works aggregate those features using the mean. We investigate five different feature aggregators: Average [58], Max [11], LogSumExp, Lp Pooling [38], and Self Attention [94].

Q4. **How to develop an indoor classification model that generalizes for the CSAM environment?** With the best model for the indoor scene classification method, in collaboration with law-enforcement agents, we evaluate our model in a real CSAM dataset. For that, we send the model with the learned weights to agents, who evaluate our method and return the results obtained.

5.3 Evaluation

We defined two pipelines for evaluation: a few-shot evaluation and a general comparison. The former follows the evaluation protocol for few-shot learning so that we can compare our work with other FSL methods. The latter compares our results with general models (non-FSL) applied to the same dataset. In both pipelines, we evaluate the methods through episodes composed of a support and a query set.

Few-Shot Evaluation

This pipeline uses only the test set. We need to build our support and query set from each evaluation episode using this set. We randomly sample N-way K-shot samples to compose the

support set. Then, we randomly select x samples per class to compose the query set, where x is a parameter that determines the number of samples per class we want to classify. The samples in the query set need to be disjointed from those in the support set.

General Evaluation

Instead of using the test set to generate both the support and the query set, we use the validation and the test set in this pipeline. From the validation set, we randomly select N-way K-shot samples to compose the support set in each episode. Then, in each episode, evaluate all the samples in the test set, which means that the query set will be composed of all the samples in this set. When we follow the few-shot evaluation, the query set in each epoch is balanced, but we can not assume that for the general evaluation.

When following this evaluation pipeline, we can compare our work with other works that do not follow the FSL paradigm. Because we use the support set to classify the samples in the query set, it would not be a fair comparison if we followed the Few-Shot Evaluation pipeline since the test set is never used for training other machine learning paradigms. So we can make a fair comparison using the general evaluation pipeline.

This pipeline does not invalidate the FSL protocol; we use only a few samples in the support set. The difference is that the support and the query sets can have samples from different domains.

5.4 Metrics

Accuracy is a metric widely used in machine learning classification problems to determine how a model performs across the classes of a dataset, that is, the percentage of correct predictions of a model. This metric calculates a ratio between the number of samples predicted correctly and the total number of samples predicted. The formula can be seen in Equation 5.1.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}. \tag{5.1}$$

The accuracy is a metric that only works if the number of samples predicted is the same for all classes. Otherwise, the value will not be representative. For example, imagine we have two classes; for one class, we have x samples, and for the other, we have $100x$ samples. Suppose the model is good at classifying the second class but could be better at classifying the first one. In that case, the accuracy in Equation 5.1 would be high, but it does not mean the model is good at classifying both classes. Therefore, when the number of samples in each class is imbalanced, it is essential to use a balanced accuracy, considering the mean of the accuracies by class (Equation 5.2).

$$\text{Balanced Accuracy} = \frac{1}{N}\sum_{1}^{N}\text{Accuracy}_N. \tag{5.2}$$

Our query set, the samples we want to classify, is balanced; we have the same number of samples for each class. Therefore, we do not need to use balanced accuracy, and the simple

accuracy is representative of our dataset. The accuracy is a good metric for understanding the average performance of the model, but it does not show how the model is performing across each class. To also understand that, we use a confusion matrix. Figure 5.2 shows how a confusion matrix works. In the main diagonal, we have all the model's correct predictions, and the other fields represent how much the model confused the correct class with other classes presented in the dataset. Both metrics will help us understand how a model performs on our dataset.

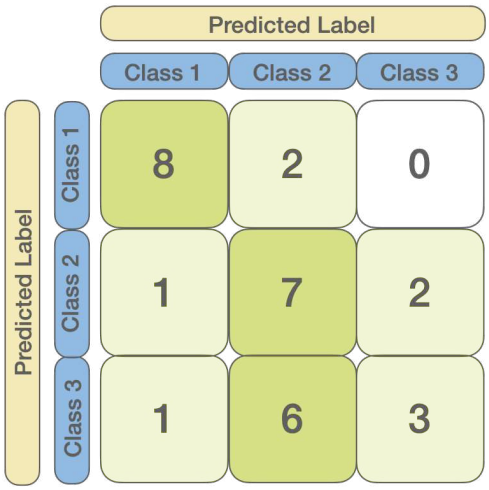


Figure 5.2: Illustration of a confusion matrix.

Chapter 6

Results and Discussion

In this chapter, we present the experiments performed and the results obtained. In Section 6.1, we present the implementation details, such as the few-shot protocol and the experimental designs. Next, in Section 6.2, we present the results obtained in our experiments, divided into four subsections. In Section 6.2.1, we compare few-shot methods. First, we compare purely convolutional methods with Transformers-based ones. Then, we compare the usage of Transformers with feature adapter and feature extractor. Based on these comparisons, we chose the best model to perform the following experiments. In Section 6.2.3, we compare different aggregation methods from the chosen model to understand the best one for our problem. Following, we present the evaluation results, divided into three sections. In Section 6.2.4, we evaluate the indoor scenes' dataset. In Section 6.2.5, we report the model's results on the OOD Scenes dataset. Last, in Section 6.2.6, we present the final results in the CSAM dataset. Figure 6.1 summarizes the experiments.

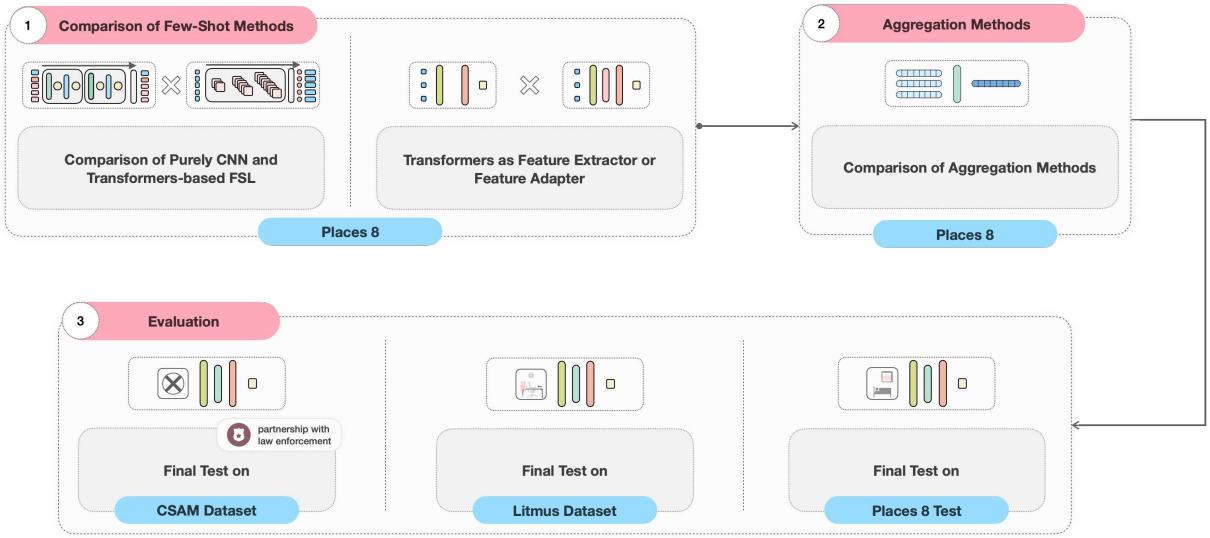


Figure 6.1: Pipeline for the experiments in this Master's thesis. In purple, we have the section name where we present the experiment's results. In aquamarine, we show the dataset used for each experiment.

6.1 Implementation Details

We followed the pipelines presented in Chapter 5 for all the experiments. In order to find the best model for our task, we performed experiments with pre-trained and fine-tuned models.

The target task is classifying indoor scene images from the classes in Places8 (see Section 4.3.1). For that, we used a few-shot protocol of 8-way 5-shot, meaning that we have only five samples per class to classify in the meta-testing stage. We applied this few-shot protocol in all the experiments performed.

We compared nine few-shot works, and all follow the few-shot protocol, but they are quite different in the model. Because of that, we did not use the same parameters for all the experiments for fine-tuning. In this case, we followed the parameters as suggested in the works. Table 6.1 details each experiment's parameters. Most of the works are pre-trained on *mini*ImageNet (Section 4.1), except for P>M>F [47], pre-trained on ImageNet-1K [24]. For fine-tuning, we used Places600 (Section 4.2).

Table 6.1: Parameters used during training for reproducing few-shot learning works. "LR" stands for learning rate.

Method	Model		Training Parameters				
	Backbone	Classifier	Initial LR	Scheduler	Optimizer	#Epochs	#Episodes
Baseline++ [19]	ResNet-18	Linear	0.001	-	Adam	400	100
ProtoNet [101]	ResNet-12	Euclidean Distance	0.001	Step LR	Adam	100	2000
RelationNet [104]	Conv-64	Custom	0.001	Step LR	Adam	100	2000
MAML [32]	ResNet-12	Linear	0.001	Step LR	Adam	100	2000
LEO [93]	Linear Encoder-Decoder	Linear	0.001	Cosine Annealing LR	Adam	100	2000
FEAT [122]	ResNet-12	Euclidean Distance	0.002	Step LR	Stochastic Gradient Descent	200	100
SSFormers [17]	ResNet-12	Cosine Similarity	0.001	Step LR	Adam	40	200
CrossTransformers [26]	ResNet-34	Euclidean Distance	0.0006	Exponential LR	Adam	40	200
P>M>F (ProtoNet) [47]	ViT	Cosine Similarity	0.00001	Step LR	Stochastic Gradient Descent	100	2000

Most of the work in this Master's thesis employed ProtoNet with ViT as the backbone. For those experiments, we trained the model for 100 steps (or epochs); in each step, we performed 2,000 episodes. The metrics are updated after each epoch.

In order to compare the works, during the testing phase, we followed the same protocol adopted in most few-shot works. We evaluate the model for 10,000 episodes, an episode being an 8-way 5-shot setup. In each episode, we have 15 samples per class to be classified, so we aim to classify 120 samples (8 classes $\times$ 15 samples). We report the mean of top-1 accuracy and 95% confidence interval in the episodes.

We used a single Nvidia Tesla T4 with 14GB to perform those experiments. The training time was close to 1 day and 20 hours, and the inference time was about 37 minutes following the few-shot testing parameters explained.

6.2 Experimental Results

We conducted a series of experiments using pre-trained backbones to extract feature vectors, and we also fine-tuned those networks on the Places600 dataset. We compared those models and chose the best one for the research testing aggregator methods. After that, we tested the model on the evaluation datasets. First, we evaluate the model on the Places8 test set. Then, to understand the model's generalization ability, we evaluate it on the OOD Scenes dataset. Lastly, through agents from the Federal Police, we evaluated our model on the CSAM and RCPD datasets.

6.2.1 Comparison of Few-Shot Methods

First, we apply existing models to our dataset, evaluating the pre-trained model and fine-tuning the model for indoor scenes using Places600. We classified the methods into purely convolutional and Transformers-based. The Transformers-based methods can be used to adapt the feature vectors (or embedding) or to extract the features themselves. Table 6.2 reports the results of FSL methods pre-trained and fine-tuned.

Table 6.2: Results of few-shot methods on Places8 validation set. We report the results of the model without fine-tuning and fine-tuning. We report top-1 accuracy and a 95% confidence interval.

Classification	Model	Backbone	Pre-trained Dataset	Pre-trained Accuracy (%)	Fine-tuning Accuracy (%)
Pure CNN	Baseline++ [19]	ResNet-18	–	–	38.36 ± 0.090
	ProtoNet [101]	ResNet-12	*mini*ImageNet	37.48 ± 0.095	43.13 ± 0.097
	RelationNet [104]	ResNet-12	*mini*ImageNet	30.49 ± 0.086	38.69 ± 0.094
	MAML [32]	Conv-64	*mini*ImageNet	34.42 ± 0.092	36.42 ± 0.095
	LEO [93]	ResNet-18	*mini*ImageNet	31.09 ± 0.093	32.66 ± 0.910
Transformers	FEAT [122]	ResNet-18	*mini*ImageNet	45.43 ± 0.090	45.34 ± 0.100
	SSFormers [17]	ResNet-12	*mini*ImageNet	41.20 ± 0.110	46.27 ± 0.120
	CrossTransformers [26]	ResNet-34	*mini*ImageNet	46.67 ± 0.095	45.07 ± 0.216
	ProtoNet (P>M) [47]	ViT Small	ImageNet-1K	68.76 ± 0.091	**71.86 ± 0.097**
		ViT Small	*mini*ImageNet	45.49 ± 0.096	52.86 ± 0.095

We could only evaluate some Transformers-based methods in Section 3.3 for different reasons. We were able to reproduce URT [65] using the same datasets used in the paper, but we could not adapt the code for our dataset due to some coding restrictions. For FewTURE [44], SUN [27], HCTransformers [43], and SMKD [62], we could not even reproduce the papers because those networks are too heavy, needing more robust GPUs than the ones we had in our research environment. HCFormers [43] and SMKD [62] used 8 Nvidia RTX 3090 for training, each one having 40GB of memory; FewTure [19] used 4 Nvidia A100, which has at least 40GB of memory each one. In this Master's thesis, we used a single Tesla T4 GPU with 14GB of memory. Also, the environment for testing CSAM is resource-constrained (in Federal Police), so we should choose models that can be reproduced in such an environment.

We performed part of their proposed work for the experiments with P>M>F [47]. As Section 3.3 states, P>M>F has three stages: pre-training, meta-training, and fine-tuning. Their

fine-tuning stage occurs during meta-testing, using part of the support set to fine-tune the model for the target task. In this Master's thesis, we do not want to use any extra information from the dataset in testing time because CSAM is too constrained. So, in all the experiments with P>M>F, we only use the first two stages of the network, i.e., pre-training and meta-training; we call this model P>M.

The authors of Baseline++ do not provide their checkpoints. So, as we could not use their pre-trained backbone, we could only train the network from scratch.

We understand that comparing ProtoNet (P>M) pre-trained on ImageNet-1K is not comparable with other methods since it was pre-trained on a much larger dataset. For that reason, we implemented an experiment that uses ProtoNet with the backbone from Supervised Masked Knowledge Distillation (SMKD) [62], as they pre-trained a ViT on *mini*ImageNet, the same network used for P>M. The difference is that their pre-training was done in a self-supervised manner.

Comparing the pure convolutional methods with those based on Transformers, the models purely convolutional performed worse. The best purely convolutional model was ProtoNet, fine-tuned using ResNet as the backbone, with an accuracy of 43.13 ± 0.097%, while the worst pre-trained Transformers-based model showed an accuracy of 45.07 ± 0.216%. Comparing it with the best results, the Purely Convolutional model performed more than 9 percentage points worse than the Transformers-based model based on ProtoNet. However, one uses ResNet-12 as the backbone, and the other uses a ViT. We believe this result is because of the self-attention property that gives weights to the patches that are more important to the classification. This experiment answers our first research question Q1, showing that Transformers-based methods outperform purely convolutional ones.

Now, comparing the Transformers-based models, we have two types of methods: the ones that use self-attention to adapt the embeddings generated from a convolutional backbone, and ProtoNet using a Transformers' architecture as the backbone (P>M). To make a fair comparison, we compare P>M pre-trained on *mini*ImageNet. We can see that P>M reaches an accuracy of 52.86 ± 0.095%, six percentage points superior to CrossTransformers, the second-best method. This result suggests that using Transformers as a feature extractor can yield great results. This answers our second research question Q2, showing that using Transformers as a feature extractor leads to better results.

This backbone must be trained using self-supervised learning to use ViT pre-trained on *mini*ImageNet. This is because ViT does not have an inductive bias like CNNs. SMKD [62] try to overcome that problem using self-supervised pre-training, following the training framework proposed on image BERT pre-training with Online Tokenizer (iBOT) [130].

We can also compare the two P>M experiments, one using a Transformer network pre-trained on *mini*ImageNet and the other pre-trained on ImageNet-1K. Not surprisingly, pre-training Transformers with more data leads to better results, suppressing results of the network pre-trained with *mini*ImageNet by more than 19 percentage points. P>M>F also reached the same conclusion, showing that using extra data in the pre-training stage reaches better results [47], but they did not compare the usage of a small dataset in the pre-training stage.

According to the authors of P>M>F [47], extra data may violate the definition of FSL. In their words, "external data violates definitions of the FSL problem that insist on a specific limited meta-train set". In this Master's thesis, we do not follow that definition. We should take

advantage of the advanced literature in other Computer Vision tasks and use it to improve FSL.

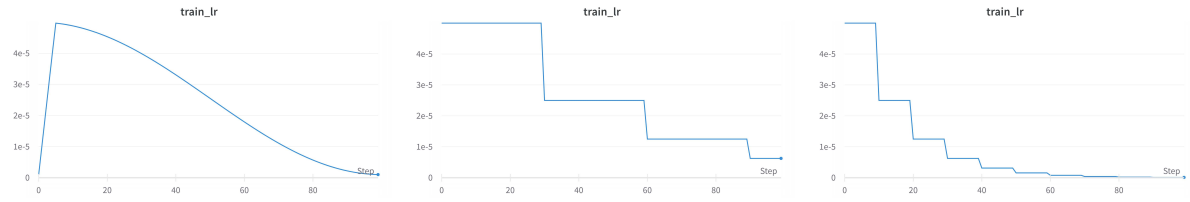
Therefore, we chose P>M pre-trained on ImageNet-1K to continue experimenting, as they led to the best result on the validation set of Places8.

6.2.2 Experiments with P>M

Given that P>M performed better than other FSL methods, we explored some changes in their network: the learning rate scheduler, the temperature scale, and two robust backbones, ViT and ResNet-50, pre-trained on a supervised and self-supervised manner.

Learning Rate Scheduler

The initial learning rate (LR) in the P>M>F meta-training stage is set to 5e-5. We followed our experiments using the same learning rate. However, we aimed to check the impact of the learning rate scheduler on the model for our task. To do that, we evaluated the cosine scheduler, the same scheduler used in the original work, and the step decay scheduler. We evaluate two decay steps for the step scheduler: 10 and 30 epochs. Figure 6.2 shows the curve for each scheduler when using cosine. We also followed P>M>F and used warm-up.



(a) Cosine w/ warmup (b) Step w/ 30 epochs decay (c) Step w/ 10 epochs decay

Figure 6.2: Curves of learning rate scheduler with an initial learning rate of 5e-5.

Table 6.3 shows the results of this experiment. All schedulers performed closely, and none can be considered better than others for our task. Our hypothesis for these results being so close is that our training set is too small. For simplicity, we continued the experiments using a step learning rate with a decay of 10 epochs.

Table 6.3: Results of the impact of learning rate scheduler on the model on Places8 validation set. We report top-1 accuracy with 95% of confidence.

Scheduler	Accuracy (%)
Cosine	71.98 ± 0.196
Step w/ 30 epochs decay	71.95 ± 0.195
Step w/ 10 epochs decay	**72.06 ± 0.196**

Temperature

The temperature (τ) is a hyperparameter that scales the logits before passing it through softmax [39], following Equation 6.1. The temperature scale can increase probabilities entropy as

$\tau \to \infty$ and decrease the entropy as $\tau \to 0$. Therefore, with small τ, the high probabilities become higher while the small probabilities become smaller. It was first proposed to be used only at test time to control the randomness of predictions. However, now, the scale is being used in the training step, so the softmax entropy impacts the loss; for cross-entropy loss, the temperature can impact the generalization of the model [2].

$$\sigma(z_i) = \frac{\exp(z_i/\tau)}{\sum_{j=1}^{N} \exp(z_j/\tau)}. \tag{6.1}$$

We aimed to understand the effect of the temperature on our results, so we considered six temperature values: 1, 0.1, 0.07, 0.03, 0.01, and P>M>F's temperature, a trainable temperature initialized as 0.1. Those experiments were done under the same parameters as the best model by now, that is, we are using P>M>F with ViT Small as the backbone. Table 6.4 shows the results of these experiments.

Table 6.4: Impact of temperature on P>M>F with ViT Small as backbone on Places8 validation set. We report top-1 accuracy and 95% confidence.

Temperature	Accuracy (%)
1	63.78 ± 0.097
0.1	71.78 ± 0.087
0.07	**72.50 ± 0.086**
0.03	72.37 ± 0.087
0.01	70.95 ± 0.089
P>M>F	71.86 ± 0.087

Although P>M>F uses a learnable temperature, fixed temperatures such as 0.1, 0.007, and 0.003 achieved competitive or outperformed P>M>F's temperature results. As the temperature of 0.007 showed the best performance and did not add any trainable parameter to the model, we chose this value to be the temperature hyperparameter of our model.

Backbone

We also wanted to understand the impact of the backbone in the model, so we tested two backbones: ViT Small, a Vision Transformer model, and ResNet-50, a fully convolutional model. We also explored the backbones with different pre-training strategies: supervised and using knowledge "DIstillation with NO labels" (DINO) [15], a self-supervised strategy. Results are in Table 6.5.

Self-supervised pre-trained backbones performed worse than the supervised strategies. These results go in the same direction observed by Kim et al. [53]. They observed that in most tasks, supervised pre-training performs better than self-supervised for domain transfer. For ResNet-50, the supervised approach outperformed self-supervised by more than 5 percentage points without fine-tuning, but when fine-tuning the model the results are more than 9 percentage points higher. ViT results for supervised and self-supervised strategies are closer, but the self-supervised approach still underperformed the supervised. So, we follow with a backbone pre-trained in a supervised manner.

Table 6.5: Results of backbone selection on Places8 validation set. We report top-1 accuracy and 95% confidence.

Model	Backbone	Pre-training Strategy	Accuracy w/o Fine-tuning (%)	Accuracy w/ Fine-tuning (%)
ProtoNet	ViT Small	Supervised	68.76 ± 0.091	**72.50 ± 0.086**
		DINO	66.02 ± 0.094	68.95 ± 0.091
	ResNet-50	Supervised	65.15 ± 0.097	70.55 ± 0.088
		DINO	59.63 ± 0.099	61.29 ± 0.095

Comparing both supervised pre-trained models, ViT Small performed better than ResNet-50. This reinforces that even for more robust models, pre-trained with larger datasets, Transformer-based approaches are a better option for our task. This experiment reinforces the answers from research question Q1 by comparing existing methods.

6.2.3 Comparison of Aggregation Methods

Most few-shot learning works that follow the ProtoNet approach – meaning to generate a prototype for each class in the support set – use the average to aggregate the feature vectors of the samples of the classes. As far as we know, no work explains why the average is the chosen aggregation method for metric-learning few-shot, so we do not know how other aggregators would perform in our task. Because this is so underexplored, we made it one of our core research questions (Q4).

To understand the impact of the aggregation method in our few-shot task, we investigated five aggregators (Section 2.2.1) commonly used in pooling operations: Average pooling, Max pooling, LogSumExp, Lp pooling, and Self-attention.

We perform the experiments using a ProtoNet with two backbones: ViT Small and ResNet-18. We used those backbones to understand the impact of the aggregators in our best model, which uses ViT Small as the backbone. However, we also aimed to understand the behavior of the aggregator in a simpler network, that is, using a small backbone, for example, a ResNet-18. We kept other network parameters the same as previously; the only changes are the backbone and the aggregation method. Results are in Table 6.6.

Table 6.6: Results for the aggregation methods on Places8 validation set. We report top-1 accuracy and 95% confidence.

Model	Backbone	Pre-trained Dataset	Aggregation Method (%)				
			Average	Max	LogSumExp	Lp Pooling	Self Attention
ProtoNet	ViT Small	ImageNet-1K	**72.50 ± 0.086**	65.80 ± 0.095	64.05 ± 0.096	55.21 ± 0.099	52.90 ± 0.133
	ResNet-18	ImageNet-1K	59.05 ± 0.096	52.61 ± 0.101	51.63 ± 0.101	50.50 ± 0.099	36.24 ± 0.124

Using average as the aggregator method led to our task's best result; it beat the second-best result by 6 percentage points. We also observed that all aggregators had the same behavior for both backbones. The best was average, followed by max, LogSumExp, Lp pooling, and then self-attention.

The performance of the self-attention aggregator was surprising. We expected it would perform better, given the results of the method on scene designer task [94], where the method was used to aggregate representation vectors instead of pooling operation. As a trainable method, it may need to be tuned more for the target task.

Moreover, to understand why average pooling was the best method, we plot the "t-Distributed Stochastic Neighbor Embedding" (t-SNE) [109] for the samples in the validation dataset with the t-SNE for the aggregated vectors of each class (Figure 6.3). We could see that the average prototype of each class is close to the cluster's center for all the classes. This indicates that the average is a good representation, given that we use the minimum distance between the sample and the class prototypes for classification. Corroborating with our findings, Xu et al. [119] investigated aggregation methods for Graph Neural Networks, concluding that the mean aggregator may perform well for tasks where the statistical and distributional information of the graph is more important than the structure and also can capture the distribution of elements in a multiset.

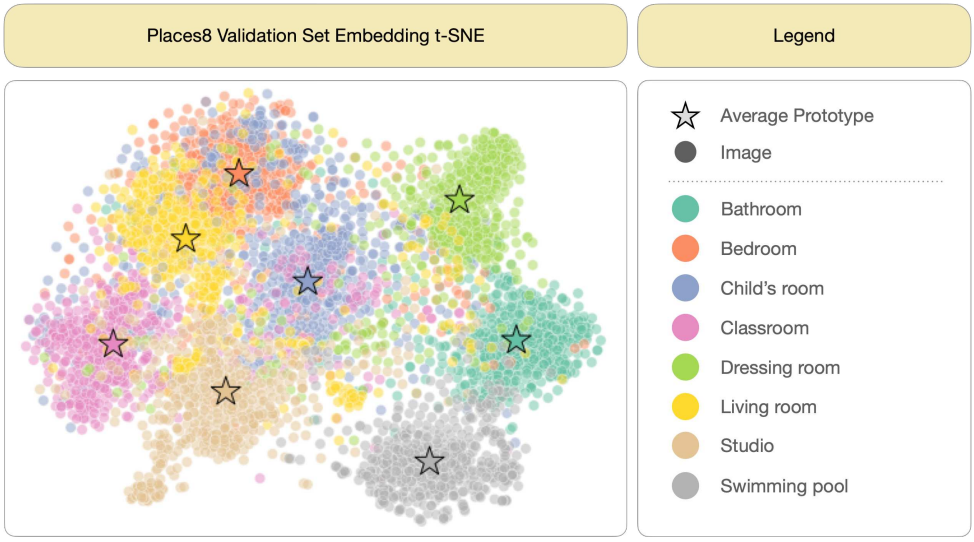


Figure 6.3: t-SNE plot of the samples from Places8 validation split (represented by the points) and the average prototype for each class (represented by stars).

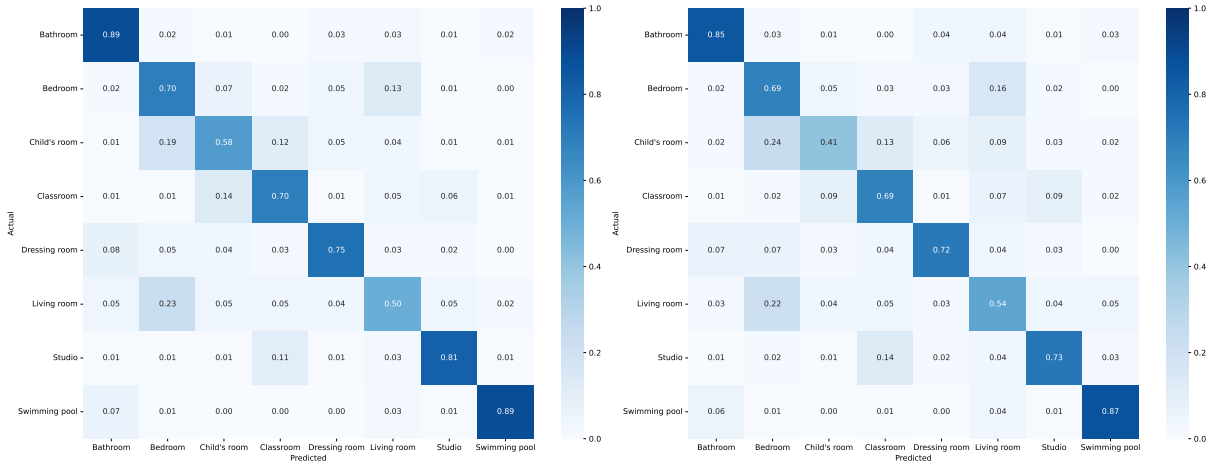
With all the decisions made, we have the best model in our hands. Our final model is a P>M, pre-trained in ImageNet-1K. Table 6.7 shows the model parameters and design choices. The final accuracy of this model on the validation set was 72.504 ± 0.086%, reporting top-1 accuracy with 95% of confidence.

Figure 6.4 exhibits the confusion matrix of the model on the validation set. We can see that the fine-tuned model is able to classify more accurately *bathroom*, *swimming pool*, and *studio*. Even though the worst prediction of the model is on *child's room*, the most desirable class for the CSAM classification, we can see that the confusion is between this class, *bedroom*, and *classroom*. We can observe the same confusion pattern for the model without fine-tuning; these confusions make sense, as sometimes it can be challenging to differentiate a regular bedroom from a children's bedroom; in some cases, the samples from *child's room* could show some kids playing, or toys with a playing table, which can be easily confused with a *classroom*. In Figure 6.5, we plot some misclassified samples by most episodes. We can see that some

Table 6.7: Final model parameters and design choices.

Model	ProtoNet
Backbone	ViT Small
Pre-trained dataset	ImageNet-1K
Initial learning rate	1e-5
Scheduler	Step decay
Decay step	10 Epochs
Decay rate	0.5
Temperature	0.07
Optimizer	Stochastic Gradient Descent
Aggregator	Average
# Epochs of fine-tuning	100
# Training episodes per epoch	2000
# Evaluation episodes	10,000

child's bedroom looks like a regular bedroom, without clear child elements, such as toys. As for the confusion with *classroom*, we also can observe that some samples seem labeled wrong, making it even more difficult for the network to differentiate the classes. The network also confused *living room* and *bedroom*, which is understandable since some samples of those classes are similar.



(a) With fine-tuning

(b) Without fine-tuning

Figure 6.4: Confusion matrix with the results of the model on Places8 validation set. The confusion matrix presents the accuracy of the predicted classes.

Comparing the model without fine-tuning (Figure 6.4b) and the fine-tuned model on Places600 (Figure 6.4a), we observed that the fine-tuned model performed better on almost all classes, except for *living room* samples. We highlight, however, the 17% gain for the *child's room* class. The overall accuracy for the model without fine-tuning was 68.76 ± 0.091%, 3.74 percentage points lower than that for the fine-tuned model. While the absolute difference is not large, the performance on *child's room* is more relevant to our use case, so we considered the fine-tuned model superior.

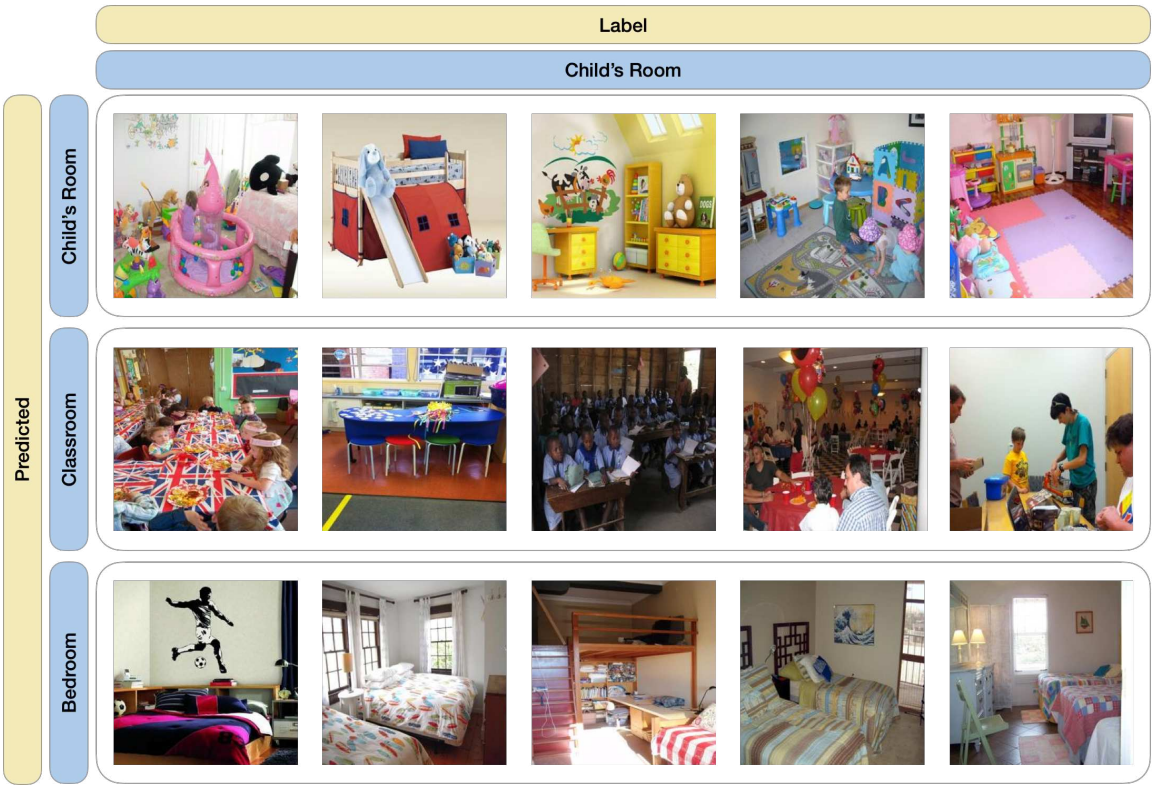


Figure 6.5: Samples from the Places8 dataset that belong to the class *child's room*, and in each row, the predicted class by most of the episodes for those samples.

6.2.4 Final Results on Places8

For the Places8 test set (Section 4.3.1), we evaluate the model in both evaluation protocols – few-shot and general – to make our model comparable for indoor scene classification.

In the few-shot protocol, we followed the few-shot learning standard evaluation protocol, as stated in the methodology (Section 5). We evaluate the method for 10,000 epochs. In each epoch, we selected 5 samples per class (following the 8-way 5-shot setting) for the support set and 15 samples per class (120 samples in total) for the query set. Those samples were randomly chosen from the test set.

For the general protocol, instead of sampling the support sample from the test set, we used the validation set, the same dataset used for making training decisions. This is not a leak since the support set is also considered training. In this protocol, we also evaluate the method for 10,000 epochs. In each epoch, we randomly sample 5 samples per class (8-way 5-shot) from the validation set to compose the support set and for the query set we used the whole test set (40,534 samples). That way, the result can be comparable with other methods that classify indoor scenes, even if the methods do not follow the few-shot protocol.

In both protocols, we report the average balanced accuracy from the 10,000 episodes and 95% of confidence. We also show the confusion matrix containing all the predicted samples from all epochs.

Figure 6.6 shows the confusion matrix for the few-shot evaluation protocol. We can observe that the confusion matrix pattern corresponds with the confusion matrix obtained from the results on the validation set, which shows us that the model is consistent in classifying indoor

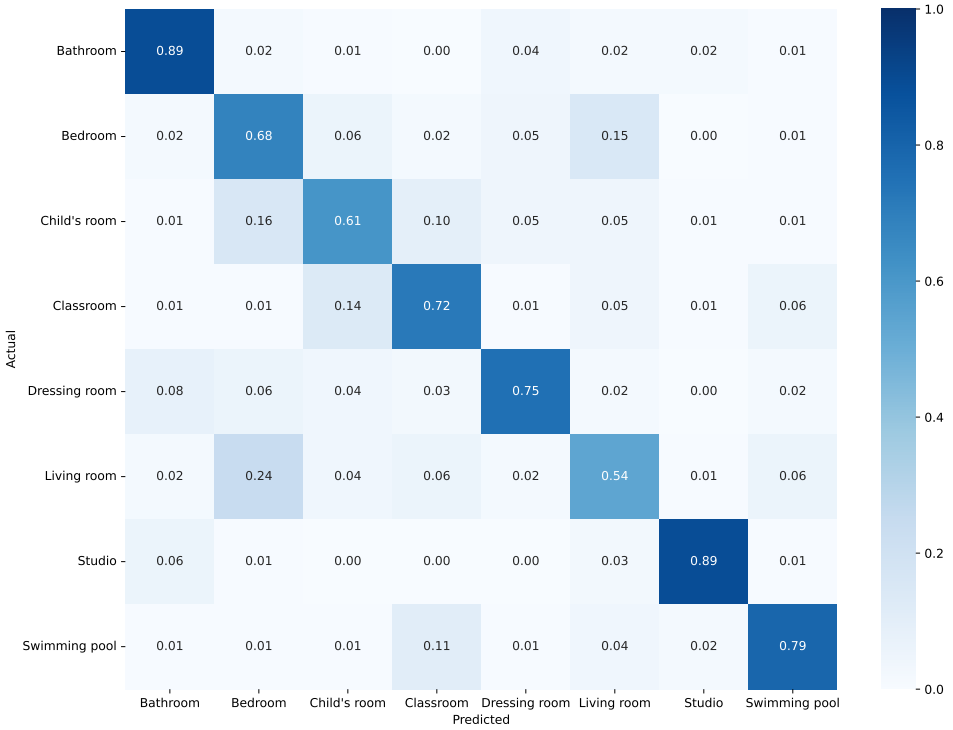


Figure 6.6: Confusion matrix with the results of the model on the Places8 test set. The confusion matrix presents the accuracy of the predicted classes following the few-shot evaluation protocol.

scenes. The main confusion was among *bedroom* with *living room* and *child's room*, also there is a huge confusion between *child's room* with *bedroom* and *classroom*. The average accuracy with 96% confidence of the model on the test set was of 73.50 ± 0.086%.

Figure 6.7 shows the confusion matrix for the general protocol. The model got more confused for *child's room*, predicting the samples as *bedroom*. The average balanced accuracy with 95% of confidence of the model following this test protocol was 69.25 ± 0.049%.

To understand the confusion between *bedroom*, *child's room*, *living room*, and *classroom*, we plot the t-SNE [109] reduction of the embeddings of the samples from the test set obtained by our backbone, as this dataset has many samples, we randomly selected a subsample of the test set of 600 samples per class to represent the dataset distribution. In Figure 6.8, we can see that the classes with confusion are overlapped. So, it is understandable that the network got confused for those classes.

Comparing our results obtained with the general evaluation protocol with the results obtained by Valois et al. [107], which is the work that proposed the Places8 dataset, and, as far as we know, the only work that reports on this dataset. Valois et al. [107] used a pre-trained self-supervised model fine-tuned on Places8. The result reported is 71.6% of balanced accuracy, only 2.35 percentage points above our results. Considering that we only used 5 samples per class to classify all the samples in each epoch, the results obtained were surprisingly good after training the model on Places600 to compare the samples.

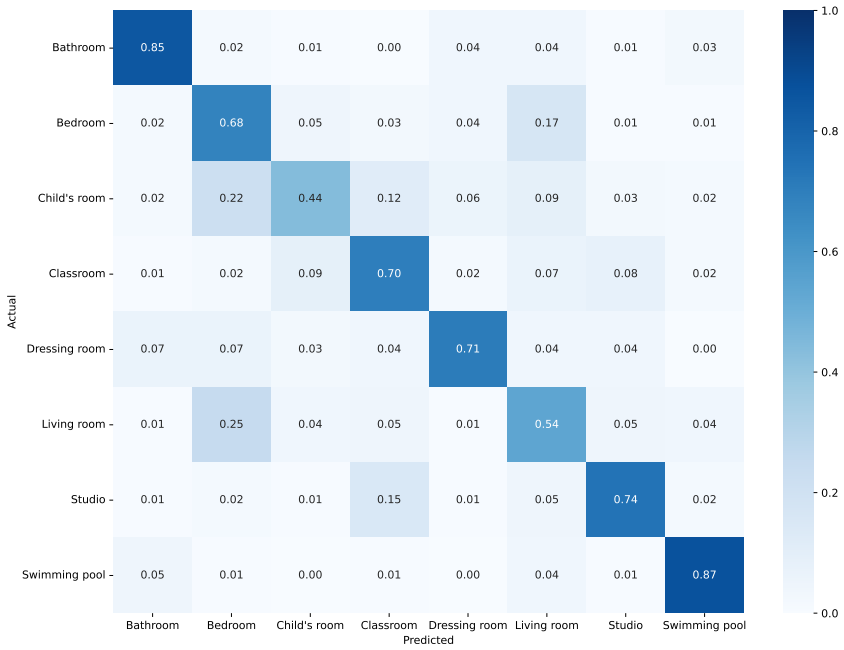


Figure 6.7: Confusion matrix with the results of the model on the Places8 test set. The confusion matrix presents the accuracy of the predicted classes following the general evaluation protocol.

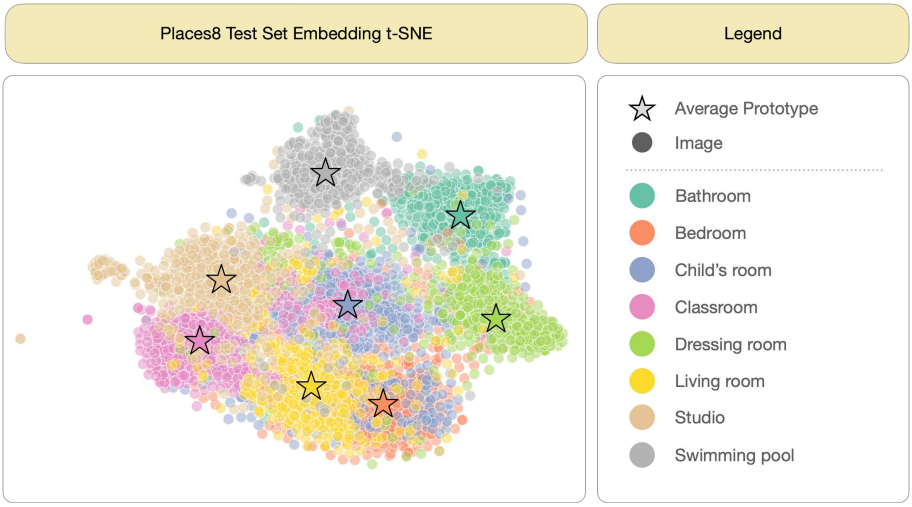


Figure 6.8: t-SNE on two dimensions for the Places8 test set.

6.2.5 OOD Scenes

Due to the small dataset size, we only followed the general evaluation pipeline in this evaluation, meaning we used the Places600 validation set to sample the support set and used all the samples in the OOD Scenes dataset as the query set (samples to be evaluated). We report the average accuracy of 10,000 epochs and 95% of confidence.

The model resulted in a 65.42 ± 0.090% of accuracy, and the accuracy by class can be seen in Figure 6.9. The confusion matrix shows that the model's behavior in this restricted dataset follows the same behavior on the Places8 validation and test dataset. That is, the model got confused on the same classes, and the most correct samples are also the same for all the datasets. For the *child's room* class, we could observe a significant drop in the performance.

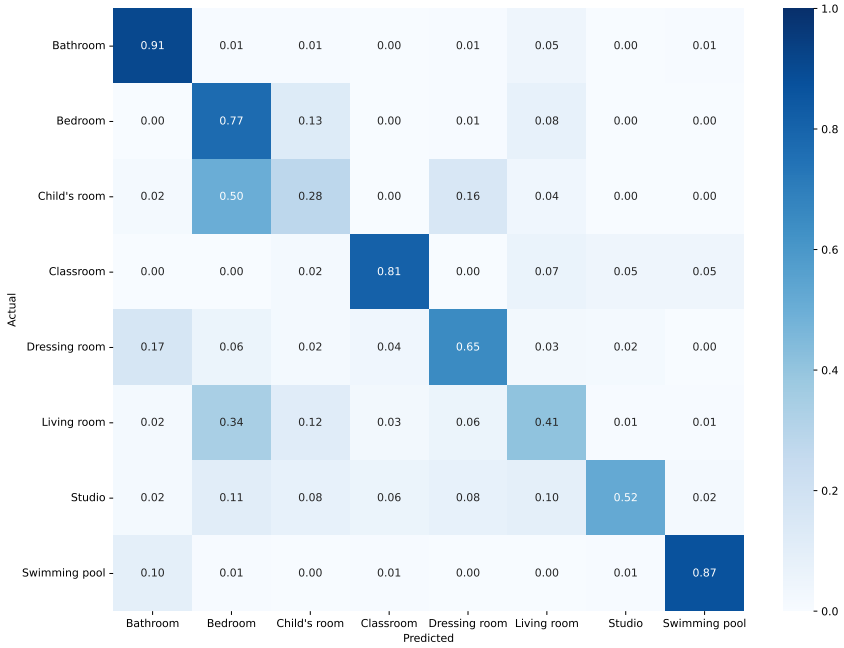


Figure 6.9: Confusion matrix with the results of the model on the OOD Scenes dataset.

Comparing this result with the results obtained by Valois et al. [107], they reported 77.5%, which is 12.08 percentage points higher than the accuracy obtained from our model.

Figure 6.10 contains the predictions through all the episodes for the OOD Scenes dataset. It shows how many episodes the model predicted the image correctly and how many episodes it was misclassified. From this figure, we can observe that most episodes get wrong 8 out of 10 images in the *child's room* subset. In the two images that most episodes got right, one has a child in a bedroom, and the other has toys on the floor. This probably happened because most of the pictures in Places600 for this class have kids and toys in their image, and in each episode, the model only have 5 samples of this class to compare and classify the query samples. For the *studio* samples, the most wrongly predicted class has a child, so this image is probably classified as *child's room*.

These predictions through all episodes also show us that the images presented in the support set are crucial for classification. Therefore, it is essential to use samples representative of the images we want to classify.

6.2.6 CSAM Final Tests

For the final tests on CSAM datasets, we collaborated with a Brazilian Federal Police agent. We developed a script for running our method in both evaluation protocols (few-shot and general). The script outputs a CSV with the predicted class for each image in all episodes. In this step, we only performed evaluation, and the training stage was done only with non-CSAM-related datasets.

In this final experimental tests, we assessed three datasets: CSAM indoor, CSAM (Section 4.3.3), and RCPD (Section 4.3.4). CSAM and RCPD are datasets labeled for CSAM classification, while CSAM indoor is a subset of the CSAM dataset but labeled with indoor scene environments.

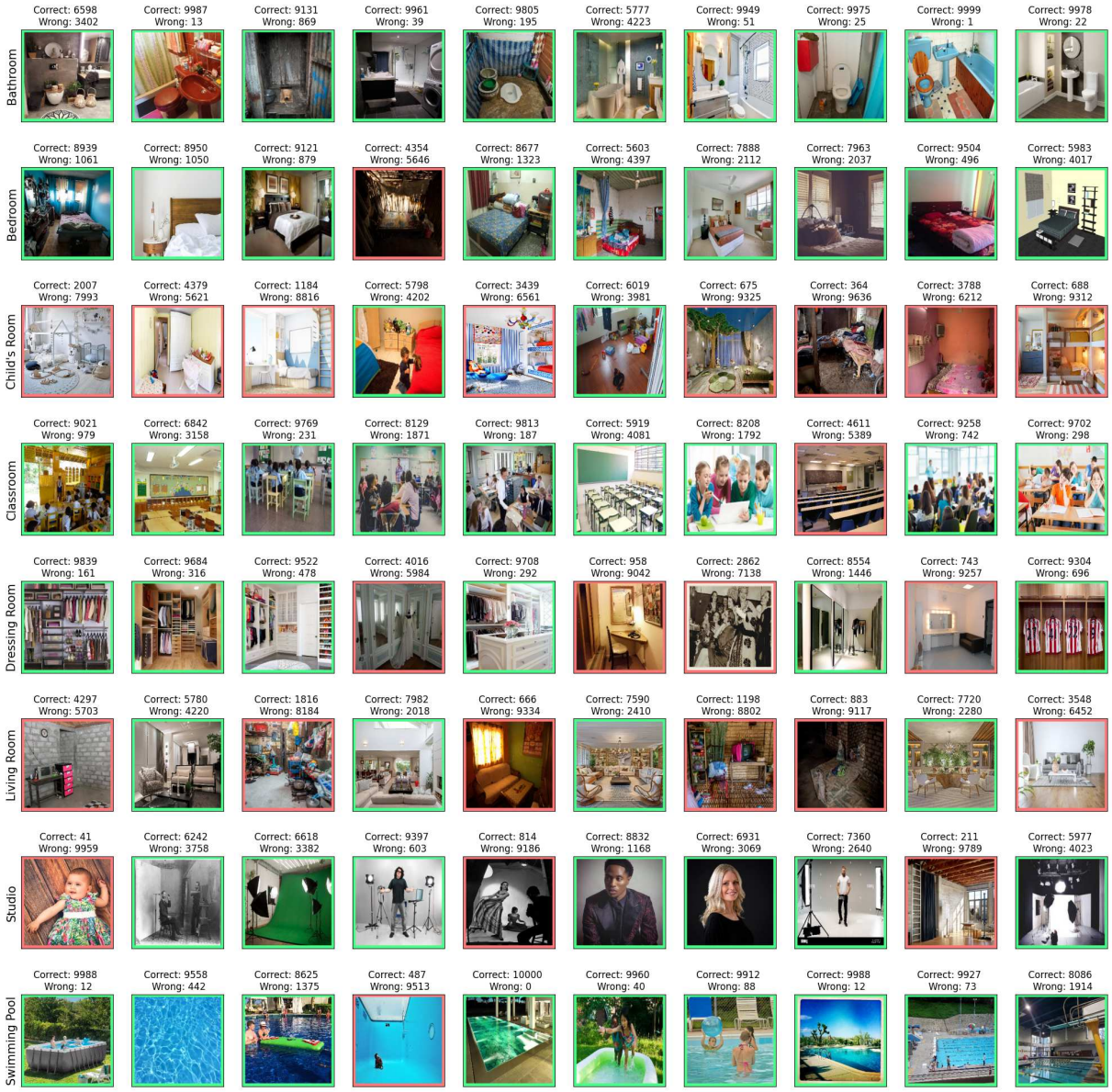


Figure 6.10: Prediction on OOD Scenes dataset through 10,000 episodes. The frame color emphasizes whether most predictions were correct (green) or wrong (red).

We emphasize that this Master's thesis focuses on few-shot indoor scene classification. However, in few-shot learning, it is possible to perform inference on classes that were not trained during training time. This way, we evaluate the final model on CSAM datasets to understand the generalization capability of our model and the importance of the indoor scene features to the CSAM classification.

CSAM Indoor

For the CSAM Indoor dataset, we performed both evaluation protocols. In both, we evaluated the model for 10,000 episodes. For the few-shot evaluation protocol, for each episode, we randomly selected 5 samples per class to compose our support set. However, instead of selecting fifteen samples per class to compose the query set, we used all the samples in the dataset that were not on the support set. The reason is that there are only eight samples for *swimming*

pool class, making it impossible for us to select fifteen samples, and other classes also have just a little annotated data. For the general protocol, in every episode, we randomly selected 5 samples per class from the Places8 validation set to compose the support set, and the query set is composed of all samples in the CSAM indoor dataset (374 samples). Figure 6.11 shows the confusion matrix for both experiments.

(a) Few-shot protocol.

(b) General protocol.

Figure 6.11: Confusion matrix with the model's results on the CSAM indoor dataset.

For the few-shot protocol, we have only six classes in the confusion matrix (Figure 6.11a) because only those classes are presented on the dataset. As we sample the classes from the datasets to compose the support set, the model can only classify the query samples into these classes since it classifies them by comparing them to the support samples. The average accuracy from the episodes with 95% confidence is 63.38 ± 0.094%. From the confusion matrix, the major confusion occurs among *bedroom*, *child's room*, and *living room*, a similar behavior

observed in the evaluation on Places8 and the OOD Scenes. The model was good at classifying *bathroom* and *studio*. Given those results, answering research question Q4, we believe the few-shot proposed model is generalized for the CSAM environment from only a few samples.

For the general protocol, we have the eight indoor classes in the confusion matrix (Figure 6.11b). However, the lines for *classroom* and *dressing room* do not have information because there is no sample in the CSAM indoor dataset classified into those classes. Nevertheless, as we are using Places8 validation to select the samples for the support set, the model can classify query samples into any of the eight indoor classes in Places8. That is why we present those classes in the confusion matrix. We can observe that the model could classify well *swimming pool* and *bathroom*; there was also a huge confusion between *bedroom* and *child's room*, which was expected given the results on Places8. The model could not classify *living room*, classifying most of the samples as *bedroom* and *child's room*. For the samples from *studio*, the model also confused with *swimming pool* and *child's room*. A similar behavior was observed by Valois et al. [107], who obtained a balanced accuracy of 36.7%. Our model achieved a balanced accuracy of 43.43 ± 0.093% with 95% confidence.

The results obtained are lower than the results obtained on Places8. This shows us a domain shift from Places8 and CSAM indoor, making it hard for the model to generalize for CSAM indoor. On the other hand, even with only five samples per class in each episode, our model performed slightly better than the self-supervised model proposed by Valois et al. [107]. This suggested that our model is less specialized for Places8 and more capable of generalization.

CSAM

The CSAM dataset consists of six classes: *CSAM*, *Suspected CSAM*, *Porn*, *People*, *Drawing*, and *Other*; none of those classes are present in Places8. Therefore, we only performed the FSL evaluation protocol on this dataset. We evaluated 10,000 episodes, where in each episode, we randomly sampled five samples per class (30 samples) to compose the support set. We randomly selected fifteen samples per class (90 samples) for the support set. Figure 6.12 shows the confusion matrix of the experiment.

From the confusion matrix, we can observe confusion among *CSAM*, *Suspected CSAM*, and *Porn*. It makes sense, given that those samples are probably visually similar, making it difficult even for police agents to differentiate those classes. The model classified the *Drawing* samples well, probably because the images from this class are from a different domain, so the samples can be easily classified. There was a small confusion between *Other* and *Drawing*, probably because the class *Other* contains images of documents, banknotes, and handwritten documents, so the model probably understood some drawings as documents and vice versa. The model was not able to classify samples from *People*; the model confused the samples from this class with *CSAM*, *Suspected CSAM*, and *Porn*. We believe that the reason for this confusion is the presence of people without clothes or not completely dressed but that do not have sexual connotations. The accuracy through all the episodes for the CSAM dataset with 95% confidence was 50.82 ± 0.109%. Considering only *CSAM* and *Suspected CSAM* classes, and mapping all the others to *Not CSAM*, the balanced accuracy increases to 53.44 ± 0.156%, almost 3 percentage points higher.

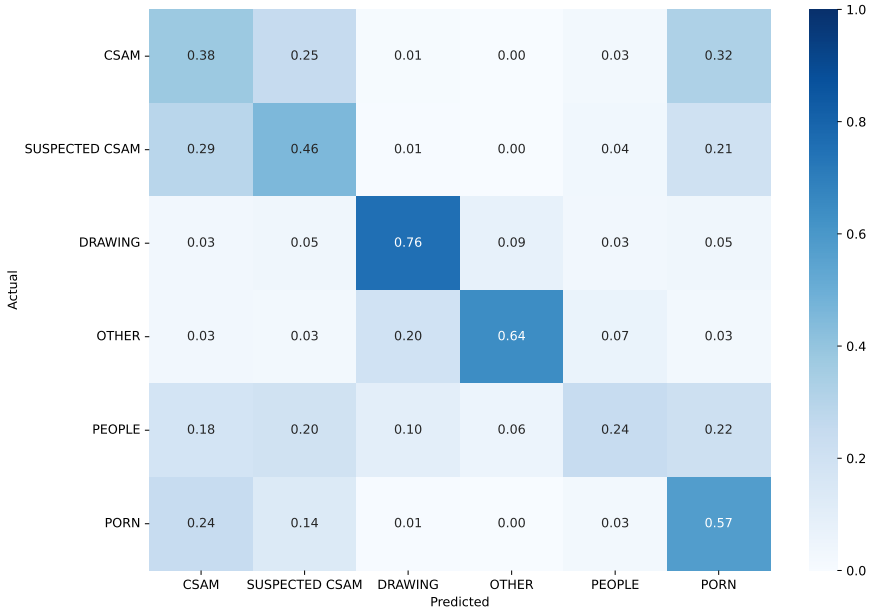


Figure 6.12: Confusion matrix with the model's results on the CSAM dataset. Values reported are the accuracy for prediction per class.

RCPD

For RCPD, we also performed only the FSL evaluation protocol since the dataset only has a binary annotation for CSAM. We also followed the same parameters for evaluation, so for each episode, we randomly sampled five samples per class to compose the support set (10 samples per episode) and 15 samples per class to compose the query set (30 samples per episode).

For this experiment, the model achieved an average accuracy with 95% confidence of 65.40 ± 0.158%. In Figure 6.13, we can see that the model was good at classifying CSAM samples, but when we look at the samples that are not CSAM, the model was not good, classifying more than half into CSAM. This behavior is probably due to the porn images contained in the category, and images containing nudity or seminudity that were not CSAM.

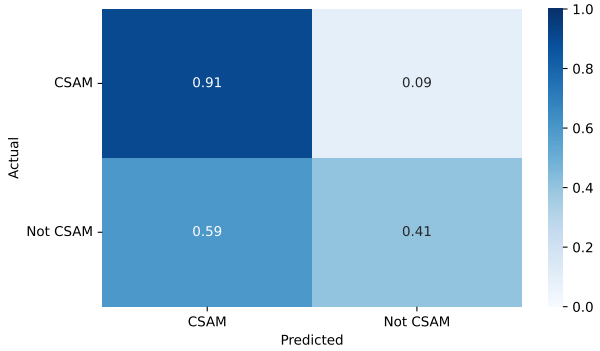


Figure 6.13: Confusion matrix with the model's results on the RCPD dataset. Values reported are the accuracy for prediction per class.

The results obtained from the experiments in CSAM and RCPD datasets showed that even with little data in the support set, the model could classify the samples quite well, given that it was trained on samples of scenes. That is indicative that it is possible to use indoor scene

classification to help in the CSAM investigation. Combined with other complementary features, it could build a robust CSAM classifier. The analysis made by Laranjeira da Silva et al. [57] also corroborates that scene features are relevant for CSAM classification.

Chapter 7

Conclusions and Future Work

This Master's thesis investigated how to improve indoor scene classification by applying metric-based few-shot learning for indoor scene classification. The objective is to help analyze Child Sexual Abuse Material, making a pre-selection of imagery of indoor scenes since these types of content are generally recorded in those environments.

We compared Transformers-based and purely convolutional few-shot learning models for indoor scenes. We analyzed the usage of Transformers-like models in two steps: in the feature extraction step or in adapting the features extracted by a convolutional neural network.

Moreover, we examined the impact of the aggregator method that generated the prototype for ProtoNet, since all the works use the average to aggregate the feature vectors. We compared several widely used pooling methods as aggregators and methods that showed promising results in a similar task.

Our results on the indoor scene dataset showed that using a Transformers-based few-shot learning method as a feature extractor leads to better results. The experiments on aggregator methods showed that using the average suppresses other methods by a large margin. The distribution of the dataset could explain this.

In the end, we evaluated the best model in the CSAM environment. The results showed that classifying indoor scenes in the CSAM environment using only a few samples can be a good triage. Also, we observed that the indoor scene is a complementary feature for CSAM classification and, combined with other features, could generate a powerful classification model.

7.1 Answers to Research Questions

The experiments performed in this Master's thesis answered the research questions as follows:

Q1. **Do Transformers-based methods outperform convolutional neural network methods for indoor classification?** Yes, for Places8, the indoor scenes dataset used for evaluation, we could observe that the methods based on Transformers were consistently better than those purely convolutional. Comparing the best purely convolutional work with the worst Transformers-based network, the latter outperformed the first by more than 2 percentage points (see Table 6.2). Also, comparing Vision Transformers and ResNet-50, a purely convolutional, used as a backbone for ProtoNet, both pre-trained on ImageNet-1K, we could observe that Vision Transform-

ers outperformed ResNet-50 by 1.95 percentage points for a supervised pre-training strategy. For a self-supervised strategy, it suppressed the convolutional backbone by 7.66 percentage points (see Table 6.5).

Q2. **What is the more accurate way to use Transformers for indoor classification? As a feature adapter or a feature extractor?** We compared four methods that are based on Transformers on Places8; three of them use Transformers as feature extractor: FEAT [122], SSFormers [17], and CrossTransformers [26], while P>M>F [47] uses a ProtoNet with Vision Transformers as backbone to extract the features. Even when trained on *mini*ImageNet, a small dataset from ImageNet [24], the ProtoNet with Vision Transformers as feature extractor outperformed SSFormers, the second-best result by more than 25 percentage points, when both models fine-tuned on Places [129]. When we only used the pre-trained models without fine-tuning, the second-best result is CrossTransformers with 46.67 ± 0.095%, while ProtoNet with Vision Transformers achieved 68.76 ± 0.091% (see Table 6.2).

Q3. **What are the impacts of feature vector aggregators on few-shot models for indoor classification?** We found that for Places8, the indoor scene dataset used for evaluation in this work, aggregating the embeddings using the average is better. We compared five aggregation methods: average, max, LogSumExp, Lp pooling, and self-attention. Average outperformed the second-best result, max aggregator, by 7.45 percentage points (see Table 6.6). We believe that the average was the best aggregator, given the distribution of the dataset. Plotting the t-SNE of the samples in Places8 and the prototype generated for each class, we could see that the prototype seems to be a good representation of the class (see Figure 6.3).

Q4. **Can we develop an indoor classification model that generalizes for the CSAM environment?** Our results showed prominent results on applying few-shot learning in the CSAM environment; with only five classes per class in the support set, the model reached a balanced accuracy of 63.38 ± 0.094% with 95% confidence. There was an expected confusion between *bedroom*, *child's room*, and *living room*, since those classes have similar visual attributes. The experiments with two CSAM datasets (CSAM and RCPD) indicated that indoor scene features are essential to classify CSAM. Combining these features with other relevant ones may lead to a significant CSAM classification, even when using only a few samples.

7.2 Challenges and Future Work

Our models performed well on few-shot indoor scene classification, given the restrictions on the number of samples and computational cost. However, some additional experiments could be done to reach better results.

First, we only performed experiments using Transformers-like architectures as feature adapters with models pre-trained on *mini*ImageNet. We could have a more fair comparison between the usage of Transformers as feature adapters or extractors if we had tested with models also pre-

trained on ImageNet-1K. Unfortunately, no work that applied Transformers-like architecture as a feature adapter trained their model on ImageNet-1K.

Most of our work was based on P>M>F [47], a ProtoNet with pre-trained Transformers on ImageNet-1K as a backbone. We explored some changes in their model pipeline and hyperparameter, but we did not investigate other classifiers, using only the cosine similarity. However, some works achieved good results using other classifiers ([17, 26, 73, 77, 96, 104]). We could try using other simple classifiers, such as Euclidean distance or linear classifier, but we could also try even more complex classifiers, like a neural network with some layers.

Moreover, our discussion on feature aggregators could be more deep. For that, it would be essential to understand the impacts of other aggregator methods ([100, 123, 124]). Some of them would have to be implemented from scratch. Also, as the average leads to good results, we could experiment with cross-attention as the aggregator. To understand the impacts of the aggregators, we could understand the data distribution of our dataset, since, depending on the distribution, some aggregators would perform better. Additionally, we could better perceive the aggregators' impact on few-shot learning in general, not only for our task, if we test the aggregator on other few-shot benchmarks.

A future direction is to employ few-shot learning generalization methods trained for indoor scene classification. Several works proposed generalization models for few-shot learning since the final task is from a different domain compared with Places, meaning the images from the final task are CSAM. We believe we could have better results applying those models in a way that makes them less sensitive to domain shift and tend to perform better in those scenarios.

We could also vary the number of samples in the support set. We used five samples per class since most works use this number of samples to compare this work with others. However, we could experiment with more samples for our application and investigate the optimal number of samples needed to compose the support set.

We could also use different metrics to evaluate our model. First, we could evaluate the model by episode, having a variation of the support and query set to understand the model's capability. Besides, as we have classes more relevant to the problem, like *bedroom* and *child's room*, and in those classes, the model lacks performance, we could investigate a way of employing restriction to those classes.

Finally, research on CSAM is challenging. Building a method that will be applied to a dataset never seen is quite difficult experimentally; not only that, but creating a classifier that will be applied to data that sometimes does not have a precise classification is also difficult. Nonetheless, we believe that the proposed work is essential to help investigations, and we encourage more researchers to follow this path. One next step in this work for CSAM would be combining more relevant features to CSAM to train the few-shot model.

Bibliography

[1] National center for missing & exploited children. https://www.missingkids.org/theissues/csam. 12

[2] A. Agarwala, J. Pennington, Y. Dauphin and S. Schoenholz. Temperature check: theory and practice for training models with softmax-cross-entropy losses. *arXiv preprint arXiv:2010.07344*, 2020. 50

[3] N. Akhtar and U. Ragavendran. Interpretation of intelligence in cnn-pooling processes: a methodological survey. *Neural computing and applications*, 32(3):879–898, 2020. 21

[4] F. Anda, N.-A. Le-Khac and M. Scanlon. Deepuage: improving underage age estimation accuracy to aid csem investigation. *Forensic Science International: Digital Investigation*, 32:300921, 2020. 12, 28

[5] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman and T. Darrell. Multi-content gan for few-shot font style transfer. In *Conference on Computer Vision and Pattern Recognition*, pages 7564–7573, 2018. 19

[6] D. Bahdanau, K. Cho and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 23

[7] P. Bateni, J. Barber, J.-W. Van de Meent and F. Wood. Enhancing few-shot image classification with unlabelled examples. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2796–2805, 2022. 22

[8] N. Bendre, H. T. Marín and P. Najafirad. Learning from few samples: A survey. *arXiv preprint arXiv:2007.15484*, 2020. 13, 19

[9] E. Bennequin. Few-shot image classification with meta-learning, 202-. https://www.sicara.ai/blog/2019-07-30-image-classification-few-shot-meta-learning. 19

[10] R. Biswas, V. González-Castro, E. Fidalgo and D. Chaves. Boosting child abuse victim identification in forensic tools with hashing techniques. *V Jornadas Nacionales de Investigación en Ciberseguridad*, 1:344–345, 2019. 12, 28

[11] Y.-L. Boureau, J. Ponce and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010. 15, 21, 42

[12] R. Brewer, B. Westlake, T. Swearingen, S. Patterson, D. Bright, A. Ross, K. Logos and D. Michalski. Advancing child sexual abuse investigations using biometrics and social network analysis. *Trends and Issues in Crime and Criminal Justice*, (668):1–16, 2023. 12, 28

[13] E. Bursztein, E. Clarke, M. DeLaune, D. M. Elifff, N. Hsu, L. Olson, J. Shehan, M. Thakur, K. Thomas and T. Bright. Rethinking the detection of child sexual abuse imagery on the internet. In *The World Wide Web Conference*, pages 2601–2607, 2019. 12, 13, 14

[14] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian and M. Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021. 24

[15] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 50

[16] D. Chaves, E. Fidalgo, E. Alegre, F. Jánez-Martino and R. Biswas. Improving age estimation in minors and young adults with occluded faces to fight against child sexual exploitation. In *VISIGRAPP (5: VISAPP)*, pages 721–729, 2020. 28

[17] H. Chen, H. Li, Y. Li and C. Chen. Sparse spatial transformers for few-shot learning. *Sci. China Inf. Sci.*, 2023. 30, 32, 33, 42, 46, 47, 64, 65

[18] W. Chen, C. Si, Z. Zhang, L. Wang, Z. Wang and T. Tan. Semantic prompt for few-shot image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23581–23591, 2023. 30, 32, 34

[19] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang and J.-B. Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 18, 30, 31, 32, 46, 47

[20] X. Chen, S. Xie and K. He. An empirical study of training self-supervised vision transformers. In *International Conference on Computer Vision*, pages 9640–9649, 2021. 24

[21] A. d. S. Correia and E. L. Colombini. Attention, please! a survey of neural attention models in deep learning. *arXiv preprint arXiv:2103.16775*, 2021. 26

[22] J. Dalins, Y. Tyshetskiy, C. Wilson, M. J. Carman and D. Boudry. Laying foundations for effective machine learning in law enforcement. majura–a labelling schema for child exploitation materials. *Digital Investigation*, 26:40–54, 2018. 28

[23] M. de Castro Polastro and P. M. da Silva Eleuterio. Nudetective: A forensic tool to help combat child pornography through automatic nudity detection. In *Workshops on Database and Expert Systems Applications*, pages 349–353, 2010. 12, 28

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 13, 33, 35, 41, 46, 64

[25] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 24

[26] C. Doersch, A. Gupta and A. Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *Advances in Neural Information Processing Systems*, volume 33, pages 21981–21993. Curran Associates, Inc., 2020. 30, 32, 33, 42, 46, 47, 64, 65

[27] B. Dong, P. Zhou, S. Yan and W. Zuo. Self-promoted supervision for few-shot transformer. In *European Conference on Computer Vision (ECCV)*, 2022. 18, 26, 30, 32, 34, 47

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 14, 24, 25, 32, 42

[29] M. Douze, A. Szlam, B. Hariharan and H. Jégou. Low-shot learning with large-scale diffusion. In *Conference on Computer Vision and Pattern Recognition*, pages 3349–3358, 2018. 20

[30] L. Fei-Fei, R. Fergus and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. 13

[31] M. Fink. Object classification from a single example utilizing class relevance metrics. *Advances in Neural Information Processing Systems*, 17:449–456, 2005. 13

[32] C. Finn, P. Abbeel and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017. 18, 31, 41, 42, 46, 47

[33] C. Finn, P. Abbeel and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 18

[34] A. Gangwar, V. González-Castro, E. Alegre and E. Fidalgo. Attm-cnn: Attention and metric learning based cnn for pornography, age and child sexual abuse (csa) detection in images. *Neurocomputing*, 445:81–104, 2021. 12, 28

[35] A. Gillespie. Legal definitions of child pornography. *Journal of Sexual Aggression*, 16 (1):19–31, 2010. 14

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 20

[37] S. N. Gowda. Synthetic sample selection for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 58–67, 2023. 27

[38] C. Gulcehre, K. Cho, R. Pascanu and Y. Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pages 530–546. Springer, 2014. 21, 42

[39] C. Guo, G. Pleiss, Y. Sun and K. Q. Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 49

[40] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu et al. A survey on visual transformer. *arXiv preprint arXiv:2012.12556*, 2020. 23

[41] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *International Conference on Computer Vision*, pages 3018–3027, 2017. 20

[42] K. He, X. Zhang, S. Ren and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 13, 42

[43] Y. He, W. Liang, D. Zhao, H.-Y. Zhou, W. Ge, Y. Yu and W. Zhang. Attribute surrogates learning and spectral tokens pooling in transformers for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9119–9129, 2022. 26, 30, 32, 34, 47

[44] M. Hiller, R. Ma, M. Harandi and T. Drummond. Rethinking generalization in few-shot classification. In A. H. Oh, A. Agarwal, D. Belgrave and K. Cho, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL `https://openreview.net/forum?id=p_g2nHlMus`. 19, 22, 30, 32, 34, 47

[45] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997. 23, 24, 31, 32

[46] T. Hospedales, A. Antoniou, P. Micaelli and A. Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44 (9):5149–5169, 2021. 17

[47] S. X. Hu, D. Li, J. Stühmer, M. Kim and T. M. Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *CVPR*, 2022. 15, 26, 30, 31, 32, 33, 41, 42, 46, 47, 48, 64, 65

[48] Y. Hu, S. Pateux and V. Gripon. Squeezing backbone feature distributions to the max for efficient few-shot learning. *Algorithms*, 15(5):147, 2022. 22

[49] Y. Hu, S. Pateux and V. Gripon. Adaptive dimension reduction and variational inference for transductive few-shot classification. In *International Conference on Artificial Intelligence and Statistics*, pages 5899–5917. PMLR, 2023. 22

[50] M. Inc. Photodna cloud services. `https://www.microsoft.com/en-us/PhotoDNA`, 2020. 12, 28

[51] A. Ishikawa, E. Bollis and S. Avila. Combating the elsagate phenomenon: Deep learning architectures for disturbing cartoons. In *2019 7th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2019. 13

[52] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer and O. Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 24

[53] D. Kim, K. Wang, S. Sclaroff and K. Saenko. A broad study of pre-training for domain generalization and adaptation. In *European Conference on Computer Vision*, pages 621–638. Springer, 2022. 50

[54] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 33

[55] J. A. Kloess, J. Woodhams, H. Whittle, T. Grant and C. E. Hamilton-Giachritsis. The challenges of identifying and classifying child sexual abuse material. *Sexual Abuse*, 31 (2):173–196, 2019. 13, 14

[56] B. M. Lake, T. D. Ullman, J. B. Tenenbaum and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017. 17

[57] C. Laranjeira da Silva, J. Macedo, S. Avila and J. dos Santos. Seeing without looking: Analysis pipeline for child sexual abuse datasets. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2189–2205, 2022. 13, 28, 62

[58] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 15, 21, 42

[59] H.-E. Lee, T. Ermakova, V. Ververis and B. Fabian. Detecting child sexual abuse material: A comprehensive survey. *Forensic Science International: Digital Investigation*, 34: 301022, 2020. 28

[60] Y. Lee and S. Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pages 2927–2936. PMLR, 2018. 18

[61] L. Leopold and H. Engelhartdt. Education and physical health trajectories in old age. evidence from the survey of health, ageing and retirement in europe (share). *International journal of public health*, 58(1):23–31, 2013. 12

[62] H. Lin, G. Han, J. Ma, S. Huang, X. Lin and S.-F. Chang. Supervised masked knowledge distillation for few-shot transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19649–19659, 2023. 19, 26, 30, 32, 34, 47, 48

[63] B. Liu, X. Wang, M. Dixit, R. Kwitt and N. Vasconcelos. Feature space transfer for data augmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 9090–9098, 2018. 19

[64] H. Liu, R. Socher and C. Xiong. Taming maml: Efficient unbiased meta-reinforcement learning. In *International conference on machine learning*, pages 4061–4071. PMLR, 2019. 18

[65] L. Liu, W. L. Hamilton, G. Long, J. Jiang and H. Larochelle. A universal representation transformer layer for few-shot image classification. In *International Conference on Learning Representations*, 2021. 30, 32, 33, 47

[66] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. Hwang and Y. Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations*, 2019. 22

[67] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 24

[68] A. López-Cifuentes, M. Escudero-Viñolo, J. Bescós and Á. García-Martín. Semantic-aware scene recognition. *Pattern Recognition*, 102:107256, 2020. 27

[69] Z. Luo, Y. Zou, J. Hoffman and L. Fei-Fei. Label efficient learning of transferable representations across domains and tasks. In *International Conference on Neural Information Processing Systems*, page 164–176, 2017. 19

[70] M.-T. Luong, H. Pham and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 23

[71] J. Macedo, F. Costa and J. A. dos Santos. A benchmark methodology for child pornography detection. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018. 12, 28, 35, 39

[72] J. Mahadeokar and G. Pesavento. Open sourcing a deep learning solution for detecting nsfw images. *Retrieved August*, 24:2018, 2016. 28

[73] N. Mishra, M. Rohaninejad, X. Chen and P. Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. 30, 31, 32, 65

[74] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein and A. Rocha. Pornography classification: The hidden clues in video space-time. *Forensic Science International*, 268:46–61, 2016. 13

[75] T. Munkhdalai and H. Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563, 2017. 20

[76] S. Narayan, A. Gupta, F. S. Khan, C. G. Snoek and L. Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *European Conference on Computer Vision*, pages 479–495, 2020. 27

[77] B. Oreshkin, P. Rodríguez López and A. Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 30, 31, 32, 65

[78] T. A. Patel, V. K. Dabhi and H. B. Prajapati. Survey on scene classification techniques. In *International Conference on Advanced Computing and Communication Systems*, pages 452–458, 2020. 27

[79] G. Patterson, C. Xu, H. Su and J. Hays. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81, 2014. 27

[80] D. Pechyony and R. El-Yaniv. *Theory and practice of transductive learning*. PhD thesis, Computer Science Department, Technion, 2009. 22

[81] C. Peersman, C. Schulze, A. Rashid, M. Brennan and C. Fischer. icop: Live forensics to reveal previously unknown criminal media on p2p networks. *Digital Investigation*, 18: 50–64, 2016. 12, 28

[82] N. Pereda, G. Guilera, M. Forns and J. Gómez-Benito. The prevalence of child sexual abuse in community and student samples: A meta-analysis. *Clinical psychology review*, 29(4):328–338, 2009. 12

[83] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein and A. Rocha. Video pornography detection through deep learning techniques and motion information. *Neurocomputing*, 230:279–293, 2017. 13

[84] F. W. Putnam. Ten-year research update review: Child sexual abuse. *Journal of the American Academy of Child & Adolescent Psychiatry*, 42(3):269–278, 2003. 12

[85] J. Qiu, Y. Yang, X. Wang and D. Tao. Scene essence. In *Conference on Computer Vision and Pattern Recognition*, pages 8322–8333, 2021. 13, 27

[86] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Conference on Computer Vision and Pattern Recognition*, pages 413–420, 2009. 27

[87] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever. Improving language understanding with unsupervised learning. 2018. 24

[88] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2016. 18

[89] L. S. F. Ribeiro and M. A. Ponti. Cross domain visual search with feature learning using multi-stream transformer-based architectures. 2023. 24

[90] J. Rondeau. *Deep Learning of Human Apparent Age for the Detection of Sexually Exploitative Imagery of Children*. University of Rhode Island, 2019. 28

[91] J. Rondeau, D. Deslauriers, T. Howard III and M. Alvarez. A deep learning framework for finding illicit images/videos of children. *Machine Vision and Applications*, 33(5):66, 2022. 28

[92] J. Rothfuss, D. Lee, I. Clavera, T. Asfour and P. Abbeel. ProMP: Proximal meta-policy search. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkxXCi0qFX`. 18

[93] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero and R. Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. 18, 31, 41, 42, 46, 47

[94] L. Sampaio Ferraz Ribeiro, T. Bui, J. Collomosse and M. Ponti. Scene designer: compositional sketch-based image retrieval with contrastive learning and an auxiliary synthesis task. *Multimedia Tools and Applications*, pages 1–23, 2022. 15, 22, 42, 52

[95] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pages 1842–1850, 2016. 18, 20

[96] V. G. Satorras and J. B. Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018. 30, 31, 32, 33, 65

[97] C. Schulze, D. Henter, D. Borth and A. Dengel. Automatic detection of csa media by multi-modal feature fusion for law enforcement support. In *Proceedings of international conference on multimedia retrieval*, pages 353–360, 2014. 12

[98] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. Feris, R. Giryes and A. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Advances in Neural Information Processing Systems*, volume 31, 2018. 20

[99] H. Seong, J. Hyun and E. Kim. Fosnet: An end-to-end trainable deep neural network for scene recognition. *IEEE Access*, 8:82066–82077, 2020. 27

[100] Z. Shi, Y. Ye and Y. Wu. Rank-based pooling for deep convolutional neural networks. *Neural Networks*, 83:21–31, 2016. 22, 65

[101] J. Snell, K. Swersky and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 15, 18, 19, 20, 21, 30, 31, 32, 42, 46, 47

[102] M. Stoltenborgh, M. H. Van Ijzendoorn, E. M. Euser and M. J. Bakermans-Kranenburg. A global perspective on child sexual abuse: Meta-analysis of prevalence around the world. *Child maltreatment*, 16(2):79–101, 2011. 12

[103] S. Sukhbaatar, J. Weston, R. Fergus et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015. 23

[104] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 15, 18, 19, 20, 30, 31, 32, 34, 46, 47, 65

[105] S. Thrun and L. Pratt. Learning to learn. In *Learning to learn*, pages 3–17. Springer, 1998. 17, 18

[106] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021. 24, 26

[107] P. H. V. Valois, J. Macedo, L. S. F. Ribeiro, J. A. dos Santos and S. Avila. Leveraging self-supervised learning for scene recognition in child sexual abuse imagery. *arXiv preprint arXiv:2403.01183*, 2024. 28, 35, 37, 38, 55, 57, 60

[108] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 14, 22, 23, 25, 32

[109] L. vd Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 52, 55

[110] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29:3630–3638, 2016. 18, 19, 20, 29, 30, 32, 35

[111] P. Vitorino, S. Avila, M. Perez and A. Rocha. Leveraging deep neural networks to fight child pornography in the age of social media. *Journal of Visual Communication and Image Representation*, 50:303–313, 2018. 12, 28

[112] P. R. R. Vitorino. Filtros especializados para a detecção de pornografia infantil. *Dissertação de Mestrado, Departamento de Engenharia Elétrica, Universidade de Brasília*, 2016. 12

[113] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang and Y. Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14408–14419, June 2023. 27

[114] Y. Wang, Q. Yao, J. T. Kwok and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):1–34, 2020. 13, 20, 29, 31, 41

[115] B. Westlake, R. Brewer, T. Swearingen, A. Ross, S. Patterson, D. Michalski, M. Hole, K. Logos, R. Frank, D. Bright et al. Developing automated methods to detect and match face and voice biometrics in child sexual abuse videos. *Trends and Issues in Crime and Criminal Justice [electronic resource]*, (648):1–15, 2022. 12, 28

[116] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang and Y. Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2018. 20

[117] Y. Xian, S. Sharma, B. Schiele and Z. Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *Conference on Computer Vision and Pattern Recognition*, pages 10275–10284, 2019. 27

[118] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. 27

[119] K. Xu, W. Hu, J. Leskovec and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018. 52

[120] W. Yan, J. Yap and G. Mori. Multi-task transfer methods to improve one-shot learning for multimedia event detection. In *British Machine Vision Conference*, pages 37–1, 2015. 20

[121] H. Yao, X. Wu, Z. Tao, Y. Li, B. Ding, R. Li and Z. Li. Automated relational meta-learning. In *International Conference on Learning Representations*, 2020. 18

[122] H.-J. Ye, H. Hu, D.-C. Zhan and F. Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 15, 30, 32, 42, 46, 47, 64

[123] D. Yu, H. Wang, P. Chen and Z. Wei. Mixed pooling for convolutional neural networks. In *Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014, Shanghai, China, October 24-26, 2014, Proceedings 9*, pages 364–375. Springer, 2014. 21, 65

[124] T. Yu, X. Li and P. Li. Fast and compact bilinear pooling by shifted random maclaurin. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3243–3251, 2021. 65

[125] Z. Yu, L. Jin and S. Gao. P^2Net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *European Conference on Computer Vision*, pages 206–222, 2020. 14

[126] A. Zafar, M. Aamir, N. Mohd Nawi, A. Arshad, S. Riaz, A. Alruban, A. K. Dutta and S. Almotairi. A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17):8643, 2022. 21

[127] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov and A. J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 33

[128] Y. Zhang, H. Tang and K. Jia. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In *European Conference on Computer Vision*, pages 233–248, 2018. 20

[129] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 27, 28, 36, 37, 64, 77

[130] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille and T. Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 48

Appendix A

Classes of Places600 Dataset

In this appendix, we present the classes filtered from Places365 [129] in the base dataset used in this Master's thesis. To have a comparable few-shot base dataset, we selected only 600 samples per class and excluded from the dataset the classes that had less than 600 samples, resulting in the classes presented in Table A.1 and Table A.2. In total, we have 268 classes, 200 in the training split and 68 in the validation split. The samples from the train split were used to train our few-shot model through the episodes. After 2,000 episodes of training, which is considered one epoch, we evaluate the model using the validation set and update the metrics.

Table A.1: Validation classes filtered from Places365, presented in the custom dataset Places600.

Validation Classes from Places600			
Airfield	Bus interior	Fountain	pond
Amusement arcade	Cafeteria	Galley	sandbox
Aqueduct	Campsite	Home office	shopfront
Arcade	Castle	Hospital	skyscraper
Archive	Chalet	Igloo	snowfield
Army base	Chemistry lab	Jail cell	soccer field
Artists loft	Cliff	Japanese garden	street
Assembly line	Clothing store	Jewelry shop	toyshop
Attic	Coast	Landfill	tundra
Auto factory	Cockpit	Marsh	utility room
Ball pit	Coffee shop	Office cubicles	village
Bamboo forest	Computer room	Orchard	vineyard
Banquet hall	Conference room	Park	volcano
Beer hall	Construction site	Patio	water tower
Boathouse	Crosswalk	Phone booth	watering hole
Bookstore	Discotheque	Playground	wet bar
Botanical garden	Downtown	Plaza	yard

Table A.2: Training classes filtered from Places365, presented in the custom dataset Places600.

Train Classes from Places600			
Airplane cabin	Courtyard	Kitchen	rainforest
Airport terminal	Creek	Lagoon	reception
Alley	Crevasse	Landing deck	repair shop
Amphitheater	Dam	Laundromat	residential neighborhood
Amusement park	Delicatessen	Lawn	restaurant
Aquarium	Department store	Lecture room	restaurant kitchen
Arch	Desert road	Legislative chamber	restaurant patio
Archaelogical excavation	Dining hall	Lighthouse	rice paddy
Art gallery	Dining room	Loading dock	river
Art school	Driveway	Lobby	rock arch
Art studio	Drugstore	Lock chamber	roof garden
Auditorium	Elevator lobby	Locker room	rope bridge
Auto showroom	Elevator shaft	Mansion	ruin
Badlands	Embassy	Manufactured home	runway
Ballroom	Engine room	Martial arts gym	sauna
Bank vault	Entrance hall	Mausoleum	schoolhouse
Bar	Excavation	Medina	science museum
Barn	Fabric store	Mezzanine	server room
Barndoor	Farm	Motel	shed
Baseball field	Fastfood restaurant	Mountain	shoe shop
Basement	Field road	Mountain path	ski resort
Beach	Fire escape	Mountain snowy	ski slope
Beach house	Fire station	Music studio	sky
Beauty salon	Fishpond	Natural history museum	slum
Beer garden	Food court	Nursing home	stable
Biology laboratory	Football field	Oast house	staircase
Boardwalk	Forest path	Ocean	storage room
Boat deck	Forest road	Office	supermarket
Bowling alley	Formal garden	Office building	sushi bar
Boxing ring	Gas station	Oilrig	swamp
Bridge	Gift shop	Operating room	swimming hole
Building facade	Glacier	Orchestra pit	throne room
Bullring	Golf course	Pagoda	ticket booth
Burial chamber	Grotto	Palace	topiary garden
Butchers shop	Harbor	Pantry	tower
Butte	Hardware store	Parking lot	train interior
Campus	Hayfield	Pasture	tree farm
Candy store	Heliport	Pavilion	tree house
Canyon	Highway	Pet shop	trench
Car interior	Hospital room	Pharmacy	valley
Carrousel	Hot spring	Physics laboratory	vegetable garden
Catacomb	House	Picnic area	veterinarians office
Cemetery	Ice cream parlor	Pier	viaduct
Clean room	Ice floe	Pizzeria	water park
Conference center	Ice shelf	Porch	waterfall
Corn field	Iceberg	Promenade	wave
Corral	Industrial area	Racecourse	wheat field
Corridor	Islet	Raceway	wind farm
Cottage	Junkyard	Raft	windmill
Courthouse	Kasbah	Railroad track	zen garden