Universidade Estadual de Campinas
Instituto de Computação

Pedro Henrique Vaz Valois

Leveraging Self-Supervised Learning for Scene Recognition in Child Sexual Abuse Imagery

Aprendizado Auto-Supervisionado para Reconhecimento de Cenas em Imagens de Abuso Sexual Infantil

CAMPINAS
2022

Pedro Henrique Vaz Valois

Leveraging Self-Supervised Learning for Scene Recognition in Child Sexual Abuse Imagery

Aprendizado Auto-Supervisionado para Reconhecimento 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 Mestre 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 Pedro Henrique Vaz Valois e orientada pela Profa. Dra. Sandra Eliza Fontes de Avila.

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Silvania Renata de Jesus Ribeiro - CRB 8/6592

Valois, Pedro Henrique Vaz, 1997-
V247L Leveraging self-supervised learning for scene recognition in child sexual abuse imagery / Pedro Henrique Vaz Valois. – Campinas, SP : [s.n.], 2022.

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. Abuso sexual na infância. 4. Visão por computador. 5. Classificação de imagem
. I. Avila, Sandra Eliza Fontes de, 1982-. II. Santos, Jeffersson Alex dos, 1984-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Aprendizado auto-supervisionado para reconhecimento de cenas em imagens de abuso sexual infantil
Palavras-chave em inglês:
Machine learning
Deep learning
Childhood sexual abuse
Computer vision
Image classification
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Sandra Eliza Fontes de Avila [Orientador]
Daniel Henriques Moreira
Claudia Maria Bauzer Medeiros
Data de defesa: 22-08-2022
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-4819-7688
- Currículo Lattes do autor: http://lattes.cnpq.br/7219990591377221

Universidade Estadual de Campinas
Instituto de Computação

Pedro Henrique Vaz Valois

Leveraging Self-Supervised Learning for Scene Recognition in Child Sexual Abuse Imagery

Aprendizado Auto-Supervisionado para Reconhecimento de Cenas em Imagens de Abuso Sexual Infantil

Banca Examinadora:

- Profa. Dra. Sandra Eliza de Fontes Avila
  Universidade Estadual de Campinas
- Prof. Dr. Daniel Henriques Moreira
  Universidade Loyola de Chicago
- Profa. Dra. Claudia Maria Bauzer Medeiros
  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, 22 de agosto de 2022

There isn't time, so brief is life.
There is only time for loving,
and but an instant, so to speak, for that.
(Mark Twain)

Agradecimentos

A pesquisa científica jamais é feita apenas por uma só pessoa, mas ela é possível graças ao apoio árduo de diversos indivíduos ao longo da trajetória.

Nesse sentido, agradeço primeiramente a minha orientadora Sandra Eliza de Fontes Avila e ao meu coorientador Jefersson Alex dos Santos. Agradeço também ao João Macedo, Thamiris Coelho, Andreza Santos, Camila Laranjeira e Gil, membros do grupo de pesquisa Unicamp/ UFMG em CSAM que sempre me guiaram nessa jornada.

Além disso, agradeço a todos os membros da Casa Azul, minha morada durante os meses que passei em Barão Geraldo. Neto, Wal, Hélio, Murillo, Isa, Bruna, Alice, Bola, Lev, Pedro, João e todos os demais que fizeram daquela casa um lar para mim.

Estendo os agradecimentos também para meus amigos do Recod, de Campinas e inclusive de São Carlos. Lucca, Wingston, Luíza, Igor, Mariana, Castello, Cia de Dança e todos que nos últimos anos me estimularam a seguir em frente.

Aliás, agradeço também ao meu orientador durante os anos da graduação, Paulino Ribeiro Villas Boas, que sempre me estimulou a tentar coisas novas e tentar aprender Python e estatística. Veja só até onde isso levou.

Não posso esquecer também dos meus amigos de Salvador. Thaís, Luíza, Sara, Gabi, Mari, Gio, Cardel, Lyra, Artur, Matheus, Bibow, Luiz, Juliana, Rodrigo, André, Pinho e todos que, mesmo longe, sempre estiveram lá quando eu precisei.

Finalmente, não posso deixar de agradecer a minha família, aos meus tios e tias, aos meus primos e primas, às minhas avós e aos meus pais, Eduardo e Virgínia, à minha irmã, Ana Luísa, e às minhas "cachorras", Bella e Suzi. Obrigado por me aguentarem todos esses anos, especialmente nesses dois últimos, tão mais difíceis para todos nós. Sem vocês tudo isso seria nada.

Resumo

O crime no século 21 é dividido em dois mundos. O mundo virtual se transformou em uma ameaça global para o bem-estar e a segurança das pessoas no mundo real. Os desafios que apresenta devem ser enfrentados com uma cooperação global unificada e devem contar mais do que nunca em ferramentas automatizadas, porém confiáveis, se quisermos combater a crescente dos crimes online. Mais de 10 milhões de denúncias de abuso sexual infantil são enviadas ao Centro Estadunidense para Crianças Desaparecidas & Exploradas todos os anos, com mais de 80% dessas de natureza primariamente virtual. Centros de investigação e instituições de acolhimento são, portanto, incapazes de processar e investigar manualmente todas as imagens com a precisão necessária. Diante disso, há a necessidade de ferramentas automatizadas confiáveis que possam trabalhar esse tipo de material com segurança e eficiência é primordial. Particularmente, o reconhecimento de cenas é a tarefa de entender os contextos do ambiente a partir de qualquer tipo de imagem. Essa tarefa é considerada útil para agrupar e classificar dados de abuso sem usar modelos treinados necessariamente em dados sensíveis. A escassez e as limitações envolvidas no trabalho com imagens de abuso sexual infantil levam ao uso do aprendizado autossupervisionado, uma nova metodologia de aprendizado de máquina que aproveita dados não rotulados para produzir representações robustas capazes de ser mais facilmente transferidas para tarefas de destino. Esta dissertação de Mestrado desenvolveu modelos de aprendizado profundo autossupervisionados pré-treinados em dados focados em cenas com 71,6% de acurácia balanceada em nossa tarefa de classificação de cenas de ambientes internos e, em média, 2,2 pontos percentuais de desempenho melhor do que uma versão totalmente supervisionada. Cooperamos com especialistas da Polícia Federal para avaliar nosso modelo em material de abuso sexual infantil e encontramos 36,7% de acurácia balanceada na classificação de cenas, mostrando que há uma lacuna entre a maneira como os ambientes são representados no conjuntos mais populares de classificação de ambientes e em material sensível.

Abstract

Crime in the 21st century is split into two worlds. The virtual world has become a global menace to people's well-being and security in the real world. The challenges it presents must be faced with unified global cooperation and must rely more than ever on automated yet trustworthy tools if we wish to combat the ever-growing nature of online offenses. Over 10 million child sexual abuse reports are submitted to the US National Center for Missing & Exploited Children every year, with more than 80% from online sources. Therefore, investigation centers and clearinghouses cannot manually process and correctly investigate all imagery. In light of that, reliable automated tools that can securely and efficiently work this material is paramount. In this sense, scene recognition is the task of understanding environment contexts from any kind of image. This task can help group and classify child sexual abuse data without requiring training on sensitive material. The scarcity and limitations involved in working with child sexual abuse images lead to self-supervised learning, a novel machine learning methodology that leverages unlabeled data to produce powerful representations that can be more easily transferred to target tasks. This Master's thesis shows that self-supervised deep learning models pretrained on scene-centric data can reach 71.6% balanced accuracy on our indoor scene classification task, and on average 2.2 percentage points better performance than a fully supervised version. We cooperate with Brazilian Federal Police experts to evaluate our indoor classification model on actual children abuse material and find 36.7% balanced accuracy on scene classification, showing a gap between the features on popular scene datasets and those depicted on sensitive material.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The virtual world has been part of society for over three decades now, and for many of us, it became an essential element of our lives once the COVID-19 lockdown had started to be enforced worldwide. In particular, our entertainment, our families, our jobs, our routines, and our privacy grew more attached to the such world, making some people question how we will ever disconnect. Since the popularization of the internet in the 1990s, the modern world has been deemed to exist without physical borders, undeniably changing how we obtain information, communicate, work, consume and socialize, but also providing an environment for misuse and abuse.

New vicious forms of crime include spreading malicious software, global networks of abuse imagery, drug trafficking, illegal organ supply chains, and even artificial intelligence (AI) enabled blackmail. As crime develops and changes in the age of connected societies, so must policing, for "the world is becoming a single jurisdiction" [60] and the efforts to fight crime in the 21st century must be made at an international level.

We emphasize that child sexual abuse networks are an ever-growing problem that has lacked proper efficient solutions for too long. In this Master's thesis, we focus on studying how to tackle the distribution of Child Sexual Abuse Material (CSAM) through AI applications. The worldwide consumption of this sort of imagery has become prevalent in the last few years, with millions of reports being received monthly by law enforcement agencies and clearinghouses worldwide. Such a phenomenon leads to the classification of thousands of related and unrelated material at each new accusation, leading to slower investigations and emotionally harmed personnel [15]. Thus, the demand for automated solutions becomes louder every year, with several different approaches to solve the CSAM classification problem. In light of the ethical and legal questions within this subject, for over a decade, hashing comparison has been the most common method for dealing with this kind of data, in which Microsoft Photo DNA became the best known CSAM hash database [76].

Given the ascension of deep learning in computer vision, many neural network solutions have attempted to aid this task. However, the barriers to working with this kind of data and the limited amount of annotations have so far seriously complicated the evolution of most initiatives. Jahankhani et al. [60] pointed out that "the complexity of the child abuse images space means that the application of the same techniques will not garner similar results", but there is room for improvement when multiple approaches are used in combination [76], meaning CSAM direct classification is not the only way to go.

As Brazil figures in the top 10 countries with the most annual CSAM reports [15], solutions able to confidently speed up the investigations are welcomed by the police force, but most importantly by the helpless children facing abuse where they should find affection.

With that in mind, scene recognition would be beneficial in speeding up investigations and finding the most probable cases to lead to an arrest [15]. For instance, detecting multiple images of a child's room potentially indicates the presence of children's images in a database, even though they are not naked or undergoing any sexually explicit activity. This approach focuses on a less abstract feature and can be intensively tested and tuned, as it serves a more general purpose than CSAM classification while simultaneously being simple to find reliable, abuse-free material for training.

In view of this scenario, we propose addressing the indoor classification problem through self-supervised learning [62], focusing on contrastive frameworks [83]. Such a methodology helps develop a more robust model and performs better than supervised transfer learning by using large amounts of unlabeled data to generate general representations that can be finetuned on downstream (target) tasks [36].

1.1 Problem Description

Deep learning has proven itself to be one of the most successful methods for creating general and scalable models using only raw data as a source of knowledge [75]. However, the recent advances in neural network architectures to produce state-of-the-art classifiers demand massive computational resources and labeled data [11, 13, 103]. Unsupervised approaches started to receive more attention in the face of the costs of annotating large amounts of data, leading to the field of self-supervised learning (SSL) [24]. The development of this area evolved fast, and several state-of-the-art models in natural language processing and computer vision are based on self-supervision [21, 62]. Many other areas that lack vast amounts of annotated data can benefit from self-supervised pretrained models, as they are especially effective in transfer learning, surpassing other supervised versions [36].

We use self-supervised learning to improve scene classification. Under this approach, we can leverage large amounts of unlabeled data to produce a more robust model for finetuning on domains other than regular transfer learning, thus surpassing supervised learning models on several tasks. SSL has been successfully applied on scene classification with SwAV [17] and SEER [42] on the Places205 dataset [134], and on scene understanding using real and synthetic images for depth measurement and object segmentation [36, 106]. However, it has yet to be tried in the indoor classification subset with the intent of helping CSAM recognition.

Whereas scene recognition may be acknowledged as a homogeneous classification task [129], indoor and outdoor environments vary greatly in textures, objects, colors, shadows, framing, and many other features. Indoor environments contain large non-textured regions that are extremely rare in outdoor environments, which can make models with impressive results for outdoors utterly fail at indoor scenes [127]. Moreover, the distribution and variety of objects are much denser in indoor environments, thus making segmentation images of these scenes more expensive and prone to error [106]. In that sense, several researchers have used synthetic datasets for indoor research, providing much more reliable depth, segmentation, and lighting

maps than real ones [107].

In the end, scene recognition is still a difficult task because one object may be essential to define the context of a whole scene [99]. This means the comprehension of an indoor scene relies on the relation between various pixel-level features, which we hypothesize to be a characteristic it has in common with the context-based nature of indecency in CSAM. The ambiguity between play and abuse can only be unraveled by techniques able to deal with dense-level information, which in turn leads us to tackle this problem with SSL.

1.2 Motivations and Challenges

The last couple of decades have seen exponential growth in the number of CSAM reports received by the US National Center for Missing & Exploited Children (NCMEC) and worldwide law enforcement agencies (LEA), reaching over 10 million in 2018 alone. The authorities point out the origin or distribution source for most abusive imagery found today to be the online collaborative platforms, such as peer-to-peer and social networks, which have grown exponentially in this century [15]. The largest hindrance to effectively combating CSAM is the lack of proper legal tools in most countries. The International Centre for Missing & Exploited Children (ICMEC) advocates that there should exist a global level of cooperation, and all countries should satisfy the five criteria below:

1. Existence of national legislation with specific regard CSAM;
2. CSAM definition;
3. Criminalization of technology-facilitated CSAM offenses;
4. Criminalization of CSAM possession, regardless of the intent to distribute;
5. Internet Service Providers' (ISPs) responsibility to report suspected CSAM to law enforcement or other mandated agency.

In 2018, 118 out of 196 countries were reported to have fulfilled at least four of the five criteria, whereas 16 countries still do not have any legislation addressing CSAM, and most punish only possession, yet allowing online visualization. Furthermore, the borderless characteristic of the internet allows for websites to be hosted in countries that do not prohibit this sort of material while it is being consumed by countries that do, making investigations extremely hard to find proof or trails of the committed crimes [60]. CSAM classification constitutes a laborious and emotionally expensive task, prone to personal bias and unintentional human subjectivity.

Thus, automatic CSAM detection has been emphasized as one of the most requested tools by agents around the world [85]. For context, the United Kingdom's government has already invested £7 million in 2020/21 to build a Child Abuse Image Database (CAID), which contains over 18.8 million indecent images of children, to help develop a classification system for investigation efficiency and reduced human contact with that kind of material [52, p. 42]. Currently, a hash comparison is the most used method for automated CSAM detection, already responsible for most of the alerts emitted today despite its limited flexibility, leading to multiple entries for the same original image on many hash databases [76]. For reports and investigations with

never-before-seen data, law enforcement personnel is still responsible for manually auditing the material, which is not only a source of bias but also a psychological strain on the agents' mental health [67].

Notwithstanding these efforts, CSAM classification presents several challenges before accurate predictions can be made automatically. As Macilotti [86] explains, the decision of whether an image is explicit enough to be considered indecent is not well defined: the notion of "sexually explicit" may sound straightforward but becomes extremely subtle more often than not, given the context of the image or the culture of the viewers. Kloess et al. [67] argue that "nudity alone is not indicative of indecency" whereas posing fully clothed can be considered erotic under some circumstances, mainly if other images depicting the same child in more abusive material is found during the investigation. From such, the most often used guidelines for classifying CSAM material, the Offense categories from the UK Sentencing Guidelines Council, shown in Table 1.1, try to encompass all these possibilities and function as a general meter of severity, but without specifying excessive detail.

Table 1.1: Offense categories from the Sentencing Guidelines Council. Categories A, B, and C represent the offense levels of the current classification system used in the United Kingdom to categorize indecent images of children according to the degree of severity depicted within them.

Level	Description
A	Images involving penetrative sexual activity, possession of images involving sexual activity with an animal or sadism
B	Possession of images involving non-penetrative sexual activity
C	Images of erotic posing

Additionally, determining the actual age of the victim is particularly challenging, and even experienced human experts have difficulty correctly defining the child's age using only images as reference [86]. Rondeau [109] illustrates that it is possible to build an accurate classifier for the *apparent age*, i.e., the age humans guess the person has, while *biological age* still lacks enough precision to be used on CSAM classification. In this sense, it is possible to reach more consistent results by segmenting through age groups, such as child and adolescent, instead of attempting to define one's age numerically [37]. On the other hand, the stiffness of the law is usually hard to tackle, for the distance between ages 17 and 18 is also the difference between guilt and innocence.

In light of the ethical and legal hurdles involved in working with CSAM, anyone outside of the police, including researchers, can absolutely never have direct access to reports or any sensitive imagery in any nation forbidding such practices, making contact with police experts a hard requirement for the study to go forward [15, 76]. Similarly, accessing a labeled CSAM database is hard, and the large ones available, such as CAID, forbid access for deep learning research or international cooperations as a security measure besides sharing the images hashes [52, p. 43].

Despite the obstacles discussed in this section, several approaches exist to assist CSAM recognition using publicly available data. Many studies have focused on detecting indecency or age measurement, which are essential yet complex features to achieve useful accuracy levels. Bursztein et al. [15] bring to discussion the fact that both review and investigation could benefit from scene clustering and recognition, with the former focusing on grouping imagery from the

same location and the latter reporting objects, environments, and landmarks different materials may have in common.

With that in mind, this Master's thesis focuses on scene classification as a means of helping CSAM investigation. As most CSAM happens indoors, experts can gather information from contextual factors to determine if children are involved in the image or even if the image depicts indecency. As Kloess et al. [67] exemplify, the background and colors can "give clues as to the type of room in which the image was taken, and whether it is likely to belong to a child or an adult" or even indicate the image was taken in "an environment that minimizes the likelihood of detection, where the victim is unaware they have been photographed".

As stated by Bursztein et al. [15], "a key challenge for law enforcement is prioritizing the reports which are most likely to lead to an arrest". Scenes like children's rooms, playrooms, bathrooms, and showers may indicate the presence of children or nudity, which can alert investigators for images most probable to be CSAM. Deep learning models that can narrow down the amount of material to be analyzed would thus be easier to develop than pure CSAM classifiers because they could use more data, receive more precise hyperparametrization, be better proof-tested, calibrated, and checked for bias and weak points.

On the other hand, the more data is available and the model is tuned, the more time and other resources are consumed. In particular, self-supervision is especially data-hungry, which makes it extremely expensive to train SSL models from the ground up [42]. Regardless, a second pretraining task is still expensive enough that researchers must carefully decide which models and data should be used. Models that use negative samples will be computationally more expensive, as the increasing number of entries requires more neural network calls. However, it is empirically accepted that pure positive models do not have the general performance of traditional contrastive learning, partly because the contrastive loss improves the variance of learning parameters when there are more negative samples [83].

Moreover, most literature on self-supervised and contrastive learning methods is fairly recent, with a large portion of results published between 2020 and 2021. Thus, one should be in close contact with the ongoing research to understand the advantages and shortcomings of the disposable models.

Additionally, the concerns with inductive biases, in general, are maximized in public SSL models. Traditionally, when training and testing machine learning models, we must assume the real-world data conforms to our dataset's distribution, which frequently fails in experiments [83]. Therefore, the lack of variety on the pretraining tasks can potentially harm the generalization and domain transferring ability these models can have, meaning that data with a higher variation of features is crucial in self-supervised pretraining [42], which we plan to deal by using large and more diverse datasets, similar to reality.

1.3 Objectives and Research Questions

1.3.1 Objectives

This Master's thesis primary goal is to seize self-supervised learning for scene classification. In this sense, we summarize our objectives as:

O1. To build indoor classification models focused on CSAM environments.

O2. To compare the performance of self-supervision pretraining versus fully-supervised approaches in the indoors recognition field.

O3. To support the CSAM investigation process with a reliable indoor classification model.

1.3.2 Research Questions

The key research questions this Master's thesis aims to answer are:

Q1. Does self-supervised outperform supervised learning for indoor classification?

Q2. Can we boost self-supervised target task performance by adding a pretraining task that uses synthetic images and their segmentation maps?

Q3. Are popular scene recognition datasets representative enough of CSAM environments that a scene classification model built from them can be used on such sensitive data?

1.4 Contributions

This Master's thesis contributes to the field of scene recognition by applying multiple SSL techniques to tackle scene classification for the first time. In this regard, we show SSL surpasses supervised learning on this task with high confidence.

Beyond that, we test our best-trained model on real CSAM and show the distribution of scenes in this data and its distinct features compared to popular scene recognition datasets.

Finally, our experiments show that combining publicly available SSL models with scene datasets improves indoor classification overall, also helping CSAM recognition and triage.

1.5 Outline

The Master's thesis is organized as follows. In Chapter 2, we review supervised, unsupervised, self-supervised, and contrastive learning frameworks under deep learning. In Chapter 3, we bring the most recent works on CSAM investigation and scene recognition. In Chapter 4, we describe the general pipelines used for pretraining and finetuning models that will answer the proposed questions and reach this research's objectives. Then, in Chapter 5, we present the used datasets, the experimental design, and its results for our target task and real CSAM data. Finally, we summarize our conclusions in Chapter 6, pointing out the contributions, challenges, and possible routes for future work.

We use the terms deep learning and neural networks interchangeably along the text.

Chapter 2

Background

Machine learning is an interdisciplinary field of statistics and computer science that builds general models exclusively from data. The field is constituted by various methods, algorithms, and procedures under which raw or processed data is used for computer vision, natural language processing, audio recognition, machine control, and many others.

In that sense, the methods that can be applied are limited by the amount of data and labels available. In general, the presence of labeled data, even in small quantities, is a hard requirement of most artificial intelligence approaches, but the high costs involved in getting millions of data points to be labeled are prohibitively expensive and time-consuming. Hence the desire to leverage raw and unlabeled data together with labeled material in the process, namely self-supervised learning.

In this chapter, we explain the different kinds of machine learning approaches in terms of the presence or absence of labeled data (Section 2.1). We focus on self-supervised learning and the contrastive learning approach, with a short review of the topic and the most important contributions. Then, we emphasize the techniques used in this Master's thesis (Section 2.2) and the optimizations essential for the obtained outcomes (Section 2.3). In the end, we briefly explain the statistical method applied to analyze the results (Section 2.4).

2.1 Types of Machine Learning

2.1.1 Supervised Learning

Supervised learning is a machine learning approach to build models exclusively from annotated data. Let X be a feature data set and Y the set of their labels. A supervised learning algorithm seeks a function $f : X \to Y$ and requires both X and Y for this task.

In classification tasks, X usually is a set of images or video frames, and Y is the class each image represents. Moreover, under the deep learning approach, the function f is an artificial neural network classifier that receives an image and outputs its class representation.

Classic supervised learning is limited by the amount of annotated data for training. Thus, this method is not scalable nor cost-effective when we consider it has to discard all unlabeled data. Even though the latter is much more abundant than the former, especially in real-world applications [62].

2.1.2 Unsupervised Learning

Unsupervised learning is a machine learning approach to build models exclusively from unlabeled data. Let X be a feature data set. An unsupervised learning algorithm seeks labeling set Y for X and a mapping $f : X \rightarrow Y$. Some algorithms might need the size of set Y as a training hyperparameter, particularly in clustering algorithms such as k-Means or Feature Agglomeration [96].

In computer vision, pure unsupervised learning algorithms have yet to show results comparable to supervised ones, especially considering that most still rely on linearly separable features or spherical distributions [122].

2.1.3 Self-Supervised Learning

Self-supervised learning (SSL) is a machine learning approach for building models using both unlabeled and labeled data. The usage of unannotated data reduces data annotation costs while also finds purpose to the massive amounts of unlabeled images and texts published daily all over the world [42]. Therefore, self-supervision follows a pattern similar to supervised transfer learning with a pretraining step, commonly called pretext task, which is designed to extract features from unlabeled data.

In short, SSL is simply a two-stage training: a pretext task using unlabeled data for a generic model pretraining and a downstream task using labeled data for specializing the model on a selected target task.

We can summarize the SSL pretext task with two goals [83]:

1. To obtain pseudo labels from the data itself, which can be produced automatically. These must be designed to teach the model general semantic features within the data set.

2. To predict or recover part of the data from other parts, where "other parts" could be transformed or corrupted data

Then, the first step in the SSL pretext task is to produce pseudo labels p_i for each entry x_i from a feature data set X. The pseudo labels are automatically generated without human annotations and can be produced with minimal cost. Thus, given n entries in X, we have a set $P = \{p_i\}_{i=0}^{n}$, and the self-supervised loss $\mathcal{L}$ is defined similarly to the supervised loss.

$$\mathcal{L}(X, P) = \min \frac{1}{n} \sum_{i=1}^{n} loss(x_i, p_i). \tag{2.1}$$

In computer vision, common pretext tasks are grayscale colorization [74], jigsaw puzzle, cutout reconstruction, image inpainting, resolution upscaling, foreground object segmentation, clustering, temporal order verification, visual-audio correspondence verification, and contrastive representation comparison (Figure 2.1). Additionally, it is crucial that such tasks can be solved from image features, which exclude hashes or metadata as viable pseudo labels [62].

Afterward, the resulting model is finetuned on specific downstream tasks, which must use a labeled dataset Z, with the difference that here it is not required to have large amounts of data (Figure 2.2). There are no limitations to the domain of the downstream task. However, it

Figure 2.1: Typical self-supervision transformations: colorization, super-resolution, and inpainting. Given the original image on the left, models are asked to recover it from various partial inputs given on the right. Figure extracted from Liu et al. [83].

has been empirically shown that ideally X and Z should be constituted of images with similar properties, such as textures, shapes, and color gamut [36].

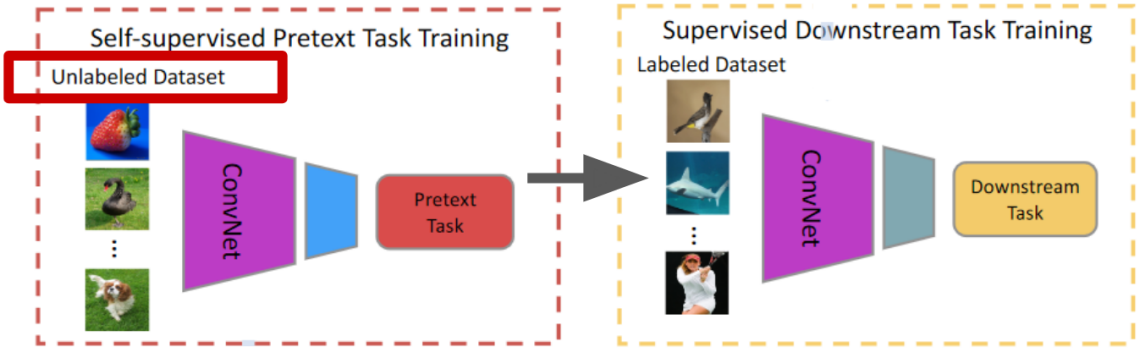


Figure 2.2: The general SSL pipeline. Simple visual features are learned through Convolutional Neural Networks (CNNs) pretraining with pretext tasks pseudo labels. Next, the learned parameters are finetuned into proper downstream tasks, whose performance measures the quality of the learned features. Figure extracted from Jing and Tian [62].

At first glance, this process might be taken as a rather complex version of supervised learning, with the addition of a new training step that requires large amounts of data most practitioners and scientists would not dispose of or would not have the resources to use. Generally, however, supervised learning produces specialized systems, oftentimes lacking out-of-domain robustness. On the other hand, SSL simplifies the usage of deep learning because it detaches the general representation learning from the specialized and application-related one [10].

Initially, we employ the pretext tasks to produce a model that learned generic image representations, able to be further improved on more specific tasks. From an interpretability standpoint, a well-trained self-supervised model should encompass highly general detectors for features such as textures, colors, and shapes for dense, pixel-level structures. On the other hand, at this point of training, the network shall not hold detectors for people, animals, objects, or places, as those would be too specialized for the pseudo labels to contain any information. Hence self-supervised models contrast with supervised versions trained on the same dataset because the former possesses more general representations than the latter, reducing the gap when they are transferred into other domains [8], i.e., when the pretrained model is trained on a new different task, e.g., pretraining in ImageNet and retraining on medical images. Moreover,

self-supervised training can use several different pretext tasks, increasing the robustness of the low-level abstractions created on the first layers of the network.

Finally, it has been demonstrated that self-supervised models outperform their supervised counterparts for ResNet-50 on image classification, detection, and segmentation in multiple benchmarks [36]. Therefore, SSL shows real advantages over supervised learning, given its higher robustness and generally better performance, especially when the downstream task has less than a hundred thousand images in the dataset [30].

Contrastive Learning

Regular self-supervision follows an unstructured application of the pretext tasks in the sense that adding a new pretext task can greatly influence the training time as it must be applied to every image in the dataset. This process cannot use multiple images to simultaneously solve one task, adding further complexity to model pretraining. Also, this can lead to limited generalization capacity of the self-supervised pretrained model because it is never shown in the context of the image, making for less useful representations.

Contrastive learning was developed as a new paradigm for self-supervised pretraining. This method focuses on creating a model that, ideally, compares two samples numerically (e.g., two images), therefore grouping similar images while pushing different ones apart. Within an unlabeled dataset, there is no prior distance measurement, and thus, for an anchor image sample, any transformed version is labeled as "similar" (or positive), and all other samples are labeled as "different" (or negative), see Figure 2.3.

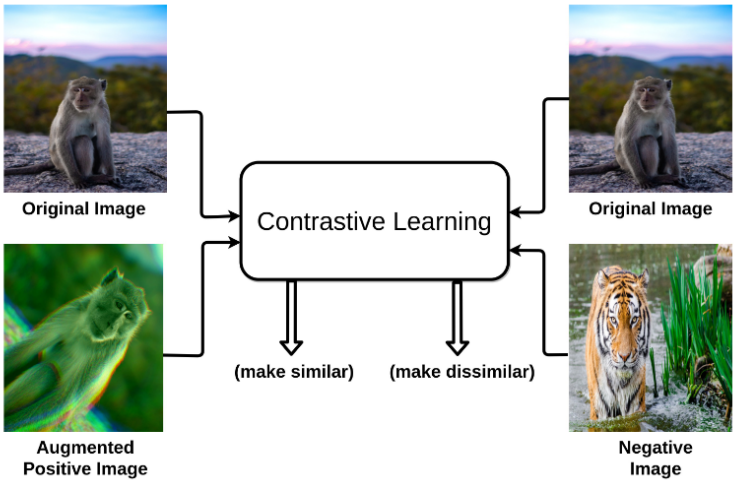


Figure 2.3: A simplified view of the general contrastive learning process. An anchor image and its augmented view are pushed closer while negative images are moved away. Figure extracted from Jaiswal et al. [61].

Contrastive learning is closely related to metric learning, an unsupervised learning subfield that aims to construct a specialized distance metric from weakly supervised data [9]. Under the deep learning umbrella, specific objective functions have been formulated to adapt metric learning to neural networks, with InfoNCE (Info Noise Contrastive Estimation) being the first one proposed [94], shown in Equation 2.2.

$$\mathcal{L} = \mathbb{E}[-\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_k e^{f(x)^T f(x_k^-)}}], \tag{2.2}$$

where x^+ is similar to x, x^- is dissimilar to x and f is an encoder.

In most published papers, the encoder is of the ResNet family [48] and the losses sometimes employ different similarity measures and weights between x, x^+ and x^-, but the framework remains the same. On the other hand, "the art of self-supervised learning primarily lies in defining proper objectives for unlabeled data" [83], and contrastive losses are one of the most studied topics with new variations being proposed after each breakthrough.

The original contrastive learning framework is the end-to-end mechanism, in which two encoders are trained simultaneously with the contrastive loss. The first methods to propose this kind of instance discrimination were IntDisc [122], CMC (Contrastive Multiview Coding) [118], and Deep InfoMax [54]. However, it was notoriously successful with Chen et al. [24], which best demonstrated the importance of hard positive samples by introducing a set of 10 data augmentations from which transformations are sampled.

Despite its state-of-the-art performance on several downstream tasks, the end-to-end training is sensitive to batch size, demanding as many as 8196 samples on the best SimCLR (Simple framework for Contrastive Learning of visual Representations) model. Therefore, on MoCo (Momentum Contrast) [50], researchers replace one of the encoders at the original contrastive learning framework for a momentum version of the other one and add a queue of negative samples to handle the large batch size employed by other methods so far. MoCov2 [26] brings some improvements over the original algorithm following the contributions laid down by SimCLR, especially regarding the hard positive sampling.

Following this path, BYOL (Bootstrapping Your Own Latents) [45] attempts to overcome the aforementioned problems regarding negative sampling by completely removing them from training. It argues that these negative views bring value only when a negative sample is similar to the anchor image, which is so rare on most datasets that it usually demands huge amounts of data per batch to find a few. To prove that statement, the researchers had to add a layer to one encoder and use an exponential moving average to update the other one, inspired by MoCo. Furthermore, the contrastive loss is the mean square error, in contrast to the usual cross-entropy function. These changes led to a better model on most downstream tasks and were more robust to smaller batch sizes than SimCLR and MoCov2.

In SimSiam (Simple Siamese) [25], this pure positive methodology is further studied but also simplified, removing the need for moving averages or momentum encoder. They found that the stop-gradient operation is the most crucial for stable learning. These changes led to a faster convergence rate, robust to even smaller batch sizes, and comparable performance.

Additionally, there is a cluster-based approach to contrastive learning. DeepCluster (Deep Clustering) was the first method of this kind to achieve competitive performance by attempting to cluster image representations as means of pretraining the network [16]. More prominently, SwAV (Swapping Assignments between multiple Views) was the model that brought clustering closer to contrastive learning by proposing a swapped prediction contrastive loss that compares representation vectors with cluster codes, which helps assign the same image's view to the same clusters. SwAV outperforms other contrastive methods on small models; it is robust to batch sizes as small as 256, and it is more computationally efficient [17]. Moreover, DeepClusterv2

was proposed together with SwAV to update the DeepCluster algorithm to more recent contrastive learning findings, and it was shown to match SwAV performance.

Ericsson et al. [36] performed a comprehensive evaluation of the aforementioned contrastive learning methods on ResNet-50 encoders and discovered that self-supervision downstream training outperforms plain supervised learning on classification, detection, depth estimation, and semantic segmentation on most Kornblith datasets [68] with the only exception being few-shot learning. None of these datasets indulge in scene recognition, but a correlation was shown between performance on ImageNet and the other datasets when the images are similar or depict the same objects, with DeepClusterv2 [16] and SwAV [17] figuring as the best algorithms overall.

On the other hand, contrastive learning has its pitfalls, and it should be considered when using pretrained models on downstream tasks. He et al. [49] found that ImageNet pretrained models have limited performance on MS COCO object detection, and Zoph et al. [135] showed it could be attributed to the gap between the instance discrimination and object detection, which limits the trustworthiness of dense classification that may depend on pixel-level information. Furthermore, contrastive methods that use SimCLR data augmentations for positive sampling tend to discard color information [36], which may be crucial for some specialized classification tasks that cannot rely on edges or texture.

Contrastive learning brings state-of-the-art models with more reliable and simpler transfer learning mechanisms, making it most suitable for projects with limited amounts of annotated data despite the differences between domains, especially in classification scenarios [83]. For reference, we show a comprehensive list of important contrastive learning publications of the past four years in Table 2.1.

2.2 Methods

2.2.1 SimCLR

The Simple Contrastive Learning of Visual Representations method (SimCLR) [24] is an end-to-end training framework for contrastive learning, which introduced composing data augmentations, nonlinear transformations in the projection head, and larger batch sizes to achieve better performance.

As illustrated in Figure 2.4, the method augments a minibatch of N images into $2N$ samples $\{\tilde{x}_i\}_{i=1}^{2N}$. For each positive pair $\tilde{x}_i$, $\tilde{x}_j$ (derived from the same original sample), the remaining $2(N-1)$ are treated as negative, and therefore the positive pair NT-Xent loss, normalized temperature-scaled cross entropy loss, becomes

$$l_{i,j} = -\log \frac{e^{sim(z_i,z_j)/\tau}}{\sum_{k=1}^{2N}[i \neq k]e^{sim(z_i,z_k)/\tau}}, \tag{2.3}$$

where $[i \neq k]$ is the Iverson bracket, τ denotes the temperature parameter and $sim(z_i, z_j)$ is the cosine similarity between the normalized representations z_i and z_j of augmented samples $\tilde{x}_i$ and $\tilde{x}_j$, respectively. Thus, the total minibatch loss is

Table 2.1: List of important contrastive learning studies ordered by published date. Acc. stands for the published ImageNet-1k top-1 accuracy. *Tiny ImageNet.

Method	Pub. Year	Characteristics	Acc. (%)	Backbone	Batch Size
DeepCluster [16]	2018	not contrastive, unsupervised clustering feature vectors, clusters with k-means ($k \approx 10,000$)	41.0	AlexNet	256
IntDisc [122]	2018	instance discrimination, data samples versus noise samples	54.0	ResNet-50	256
DeepInfoMax [54]	2019	maximizes mutual information, instance discrimination	*38.1	ResNet-50	4096
CMC [118]	2019	multiple views (color channel, depth, segmentation), instance contrastive, memory bank	42.8	AlexNet	4096
MoCo [50]	2019	instances kept in queued memory bank, momentum encoder to improve the representation consistency between the current and earlier seen keys	60.6	ResNet-50	256
SimCLR [24]	2020	instance contrastive method, two equal encoders, combined augmentations	76.7	ResNet-50	2048
MoCov2 [26]	2020	improves MoCov1 inspired by SimCLR: MLP projection head and more data augmentations	67.5	ResNet-50	256
BYOL [45]	2020	pure positive contrastive: no negative instance discrimination; online and target encoders with different sizes, applies moving average and 3 phases representation, projection, prediction	74.3	ResNet-50	4096
SimSiam [25]	2020	one network, each step two results, swapped loss, stop-gradient to prevent collapse, same image - no negative pairs	68.1	ResNet-50	512
SwAV [17]	2020	clustering contrastive method, online learning, latent variable, clustering	75.3	ResNet-50	256
DeepClusterv2 [17]	2020	improves DeepClusterv1 with ResNet-50, better augmentations, cosine learning rate schedule, MLP projection head, use of centroids	75.2	ResNet-50	256
Barlow Twins [128]	2021	batch contrastive, low negative size, low batch size, benefits from high-dim representations, cross-correlation loss, maximizes mutual information	71.8	ResNet-50	2048
MoCov3 [27]	2021	introduces vision transformer architecture in MoCo	79.9	ViT-B	1024
DINO [18]	2021	student-teacher architecture, vision transformers	80.1	DeiT-S/16	1024
SEER [42]	2021	SwAV on 1B Instagram Images for 1 epoch, RegNetY Scalable Architecture	84.2	RegNetY-256	8704
Data2Vec [6]	2022	multi-model general representation (deals with text, audio, and vision), applies vision transformer architecture	84.2	ViT-B	2048

$$\mathcal{L} = \mathbb{E}[l] = \frac{1}{2N} \sum_{i=1}^{N} l_{2i-1,2i} + l_{2i,2i-1}. \tag{2.4}$$

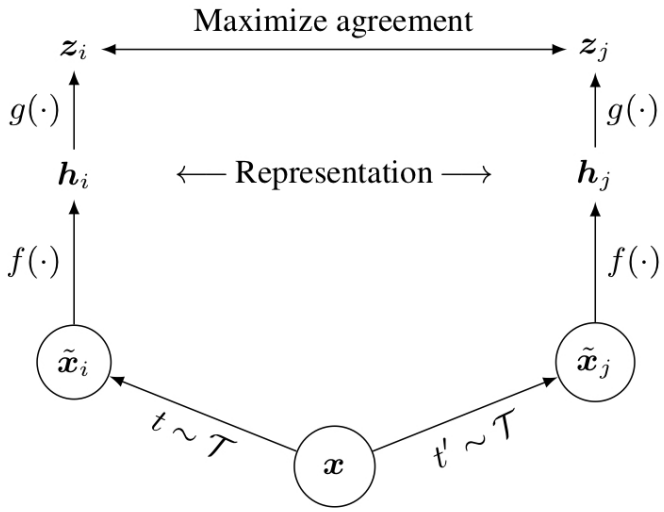


Figure 2.4: SimCLR framework: two data augmentation operators are sampled from the same distribution ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each sample to obtain a positive pair. A base encoder network f and a projection head g are trained to maximize agreement using a contrastive loss. f is the output of this self-supervised pretraining algorithm. Figure extracted from Chen et al. [24].

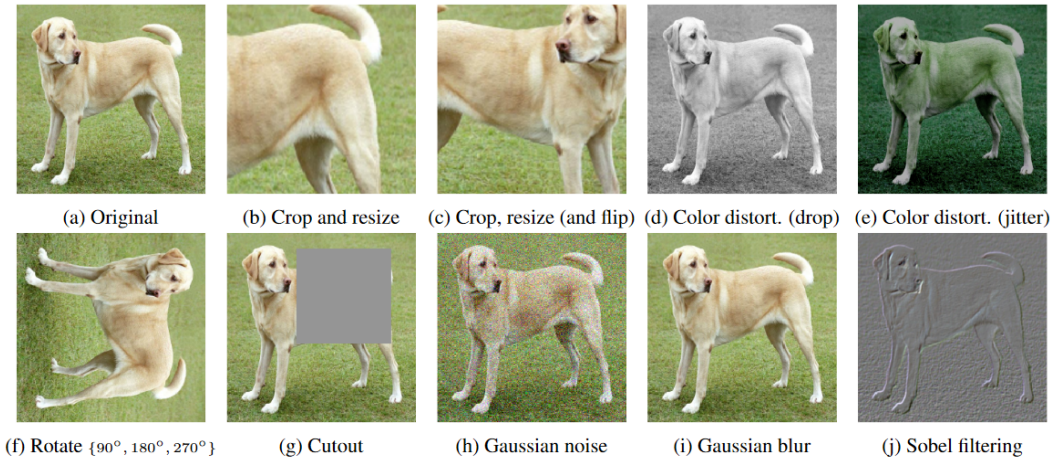


Figure 2.5: Isolated views of the transformations proposed for SimCLR yet commonly used on other contrastive learning applications. Figure extracted from Chen et al. [24].

While self-supervised learning makes intensive data augmentations on pretext tasks a systematic way to generate pseudo labels, several transformations have been proposed with varied usage throughout the literature [62]. In that regard, SimCLR introduced a set of data augmentation operations that other contrastive learning techniques have widely adopted [17, 25, 45, 128], namely crop, resizing, flip, color jitter, color drop, 90º rotations, cutout, noise, blur, and Sobel filtering, as shown in Figure 2.5.

These augmentations are randomly sampled to generate positive pairs on the contrastive framework, but it is crucial to notice that they act as a source of invariance, meaning that the contrastive loss will teach the model that it should treat an image and any of its positive pairs evenly. For instance, if the positive pair is under a grayscale transformation, it might lead the model to discard color information entirely, which has been verified for most contrastive learning pretrained models [36], but is potentially undesired for many applications that do not wish such an invariant. One possible solution is to execute the second round of SSL pretraining with a domain-specific dataset under a curated set of transformations.

One of the limitations of the SimCLR technique is the huge dependency on batch size, requiring at least 2048 images on each minibatch to show competitive performance with supervised frameworks. The larger batch sizes provide more negative examples, increasing the probability of similar negative pairs, and facilitating convergence with fewer epochs [24].

Finally, SimCLR is a simple yet effective contrastive learning technique, making it quite useful to check the self-supervision concept on new tasks and domains, even though it is limited by the maximum batch size the training infrastructure can withstand.

2.2.2 SwAV

This contrastive learning technique of Swapping Assignments between multiple Views of the same image (SwAV) [17] introduced a scalable online clustering loss and multi-crop strategy to self-supervised learning. This method does not require pairwise comparisons between views and instead compares image representations with clustering codes, which makes it more memory efficient as it can be trained with much smaller batch sizes than SimCLR [24].

The method starts from a multi-crop input, which uses two crops of predefined resolution and sample V extra low-resolution crops. For example, 2×224 pixels + 6×96 pixels, resulting in a total of 8 square views. Thereafter, the method augments a minibatch of N images into $(V+2)N$ samples $\{\tilde{x}_i\}_{i=1}^{(V+2)N}$. Next, a code q_i is computed from the representation $z_i = f_\theta(\tilde{x}_i)/||f_\theta(\tilde{x}_i)||$ by mapping a set of K trainable prototype vectors $\{\vec{c}_i\}_{i=1}^{K}$, where we denote C as the matrix whose columns are c_i. This process is shown in Figure 2.6.

$$q_i = \text{sinkhorn}(z_i C), \tag{2.5}$$

where sinkhorn is a function that executes the sinkhorn-knopp algorithm on the matrix multiplication of z_i and C to obtain the code q_i.

Then, the loss between one representation z_t and code q_s is calculated as a dot product between the code q_s and p_t, the log of a softmax of the dot products of z_t and all prototypes in C.

$$l(z_t, q_s) = \vec{q_s} \cdot \vec{p_t} \tag{2.6}$$

$$p_t[k] = -\log \frac{e^{\frac{1}{\tau}\vec{z_t}\cdot\vec{c_k}}}{\sum_{k'} e^{\frac{1}{\tau}\vec{z_t}\cdot\vec{c_{k'}}}} \tag{2.7}$$

From such, we compute the swapped loss L as

$$L(z_1, z_2, ..., z_{V+2}) = \sum_{i=1}^{2} \sum_{v=1}^{V+2} [v \neq i] l(z_v, q_i) \tag{2.8}$$

$$= l(z_1, q_2) + l(z_2, q_1) + \sum_{i=1}^{2} \sum_{v=3}^{V+2} l(z_v, q_i), \tag{2.9}$$

where we made the swapped part explicit in Equation 2.9. Notice the codes are not computed from low-resolution crops, which has been shown to increase the computational time without improving performance [17].

Finally, this loss function is averaged over all the images in the minibatch and is minimized with respect to C and parameters θ of f_θ.

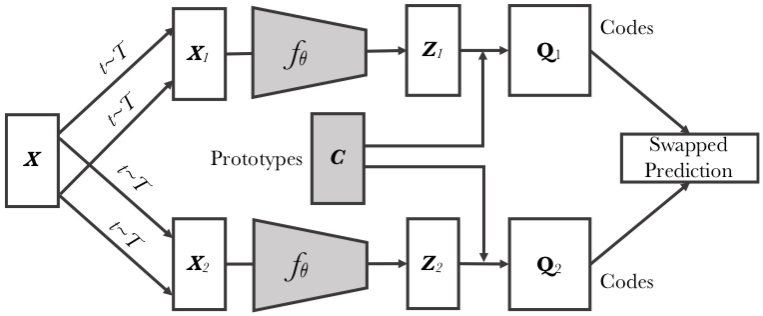


Figure 2.6: SwAV framework: two data augmentation operators are sampled from the same distribution ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each sample to obtain a positive pair. A base encoder network f_θ outputs features z_1 and z_2 for each pair and codes q_1 and q_2 are then produced by assigning features to learnable prototype vectors. The system is trained to solve a swapped prediction problem, i.e., maximize agreement between the pairs (z_1, q_2) and (z_2, q_1). Figure extracted from Caron et al. [17].

Although it requires a clustering procedure to be trained alongside the neural network, SwAV shows significant gains in performance over SimCLR, and other SSL techniques [17]. In particular, SwAV can train efficiently with a batch size of 256, while other contrastive methods typically require at least four times as much to get comparable results.

Besides that, SwAV demands less data synchronization between GPUs in multi-node or multi-GPU distributed training than SimCLR, which makes it more efficient in these cases. The multi-crop strategy is also important for this method, increasing top-1 accuracy on ImageNet by 4% on SwAV and increasing by 2% on SimCLR [17].

All in all, SwAV uses intricate clustering to build a less expensive contrastive method with even better performance when compared to other SSL techniques.

2.2.3 Barlow Twins

The Barlow Twins SSL method [7], named after Horace Basil Barlow, follows the hypothesis that sensory processing algorithms intend to transform highly redundant inputs into a factorial

code, which is made only of statistically independent components. This technique aims to simplify contrastive learning algorithms while keeping competitive performance.

Following Figure 2.7, the method augments a minibatch of N images into $2N$ samples $\{Y_b^A, Y_b^B\}_{b=1}^N$, where b is the image index in the minibatch and A, B are the two augmented versions of the same image. Then, the batch cross-correlation matrix $\mathcal{C}$ is defined in terms of the normalized representations $z_{b,i}^A$ and $z_{b,j}^B$

$$\mathcal{C}_{ij} = \frac{\sum_{b=1}^{N} z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_b (z_{b,i}^A)^2}\sqrt{\sum_b (z_{b,j}^B)^2}}. \tag{2.10}$$

Notice $\mathcal{C}$ is the cross-correlation matrix for each two coordinates in the embedding space for the whole batch, meaning that $\mathcal{C}$ is calculated once per minibatch.

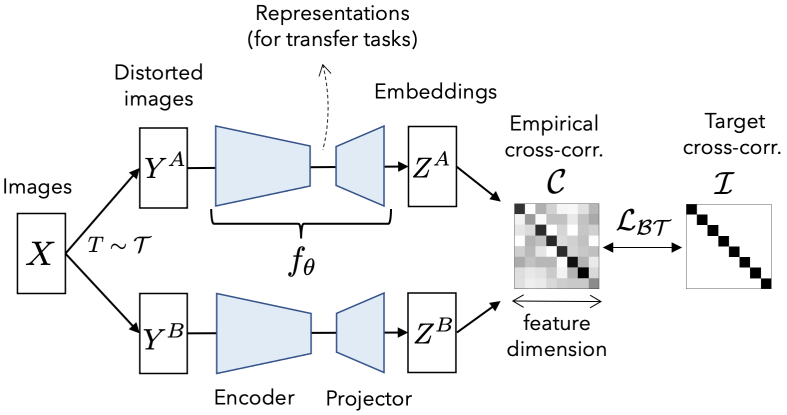


Figure 2.7: Barlow Twins framework: a positive pair is obtained from two distorted versions of an image that had transformations from a set $\mathcal{T}$ applied to it. A base encoder-projector network f_θ then outputs features z for each pair. The system measures the joint cross-correlation matrix for all embedding pairs in a batch and tries to make this matrix as close to identity as possible. Figure extracted from Zbontar et al. [128].

Hereafter, the goal is to train the model to produce $\mathcal{C} = \mathbb{I}$. In other terms, we must have $\mathcal{C}_{ii} = 1$ and $\mathcal{C}_{i\neq j} = 0$, which suggests a loss of the form

$$\mathcal{L} = \sum_i (\mathcal{C}_{ii} - 1)^2 + \lambda \sum_i \sum_{i\neq j} (\mathcal{C}_{ij})^2, \tag{2.11}$$

where $\lambda \in \mathbb{R}^+$ is a constant trade-off of the importance between the first and second terms.

This loss is composed of two terms: the invariance term $\sum_i (\mathcal{C}_{ii} - 1)^2$ tries to equate $\mathcal{C}$ main diagonal to 1, which means it intends to make the embedding invariant to transformations; the redundancy reduction term $\sum_i \sum_{i\neq j} (\mathcal{C}_{ij})^2$, on the other hand, tries to get the off-diagonal elements to 0, an attempt to decorrelate the different embedding vector components and therefore reduce redundancy. In the end, this formula is a sum of strictly positive terms, so if it reaches the minimum value of zero, we must have $\mathcal{C} = \mathbb{I}$.

In conclusion, Barlow Twins introduces a novel yet simple approach to contrastive learning, with a completely new loss while being robust to small batch sizes and without relying on clustering, which can potentially collapse into trivial solutions. Moreover, Barlow Twins

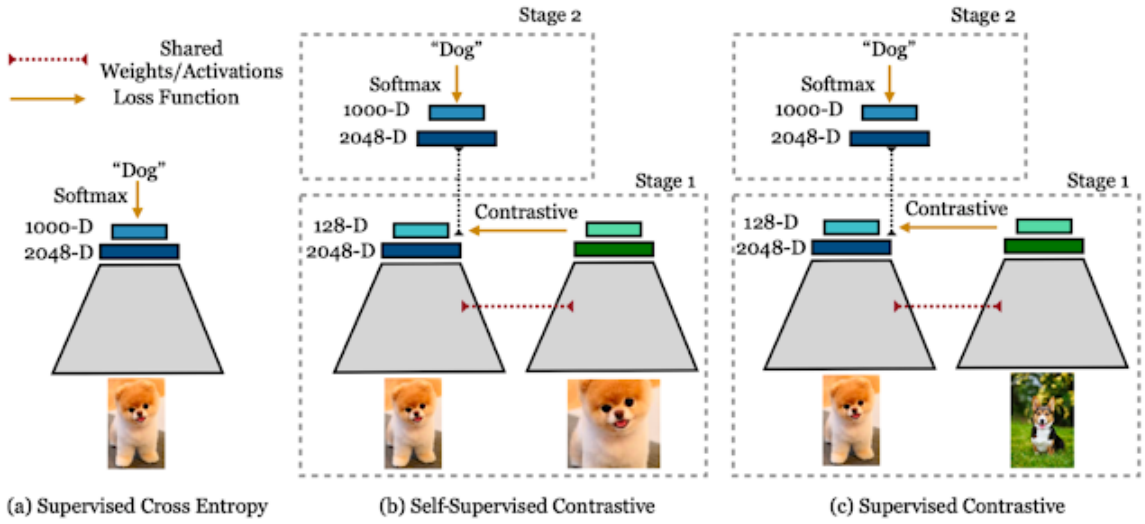


Figure 2.8: Comparison between fully-supervised cross-entropy loss, self-supervised contrastive loss, and supervised contrastive loss. (a) The fully-supervised cross-entropy loss uses labels and a softmax loss to train a classifier directly. (b) The self-supervised contrastive loss uses contrastive loss and data augmentations to learn representations without previous label knowledge. (c) The supervised contrastive loss also learns representations using a contrastive loss, but uses label information to group positives in addition to augmentations of the same image. Figure extracted from Khosla et al. [65].

greatly benefit from larger embedding dimensions, whereas other methods quickly saturate in performance [128].

2.2.4 Supervised Contrastive Learning

Supervised Contrastive Learning (SupCon) [65] borrows techniques from contrastive learning, particularly SimCLR, to perform a pretraining step leveraging known image labels. It extends the SSL approach to the fully-supervised setting by pulling together representations of images belonging to the same class while pushing apart representations of different classes.

As illustrated in Figure 2.8, the method augments a minibatch of N images into $2N$ samples $\{\tilde{x}_i\}_{i=1}^{2N}$ and labels $\{\tilde{y}_i\}_{i=1}^{2N}$. Then, we define the positive set with respect to sample i as $P(i) = \{p : \hat{y}_p = y_i, p \neq i\}$ and normalized representations $\vec{z}_i$ which is the set of indices which share the same label as i in the augmented minibatch, while the remaining are treated as negatives.

The knowledge of $P(i)$ permits an extension of the NT-Xent loss, introduced by SimCLR [24], to a version that is averaged over all positives.

$$\mathcal{L} = -\sum_{i=1}^{2N} \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{e^{\frac{1}{\tau}\vec{z}_i \cdot \vec{z}_p}}{\sum_{j \neq i} e^{\frac{1}{\tau}\vec{z}_i \cdot \vec{z}_j}}. \tag{2.12}$$

Since the SupCon loss is a direct extension of the NT-Xent loss, it shares many of its properties, advantages, and shortcomings. In particular, it relies on large batch sizes, requiring over 4096 images per batch to achieve good performance. Moreover, as with fully-supervised learning, the quality of the representations will depend on the labeling quality, meaning that it might

suffer in weekly supervised scenarios [65].

Overall, SupCon has shown itself better than fully-supervised and even SSL in some cases, although requiring larger networks like ResNet-200 to beat the latter on the popular benchmarks [65].

2.3 Optimizations

Modern deep learning algorithms train models with millions or billions of parameters, requiring massive computing power to finish in a viable time. Parallel to that, many optimizations have happened on hardware and software to enable faster deep neural network (DNN) training with no loss in performance. In this section, we explain the deep learning optimizations used in this Master's thesis to make training feasible.

2.3.1 Automatic Mixed Precision

Deep Learning traditionally uses IEEE single-precision (FP32) [64, 66, 91], but it is possible to optimize the whole procedure by using half precision (FP16) without performance loss. Automatic Mixed Precision (AMP) is the combined use of different numerical precisions in a computational method [2].

The effectiveness of AMP is a feature of modern GPU architectures[1], which can process FP16 operations much faster than FP32 with less memory consumption. Numerically, we have

1. 1/2× memory consumption
2. 2× memory throughput
3. up to 48× compute speedup[2]

However, all this speedup is not directly transferred to deep learning, and some operations must still be kept in FP32 to prevent numeric losses. In simple terms, FP16 precision is enough to process convolutions, general matrix multiplications (GEMMs), and most pointwise transformations (e.g., ReLU). On the other hand, FP32 must be used on all accumulated procedures, weight updates, and non-linear functions (losses, softmax, normalization) [90].

In the end, the final speedup and memory savings depend on the DNN architecture used, but it is certain to give at least 3.3× compute speedup on the most popular ones without any hyperparameter change, or performance loss [90].

2.3.2 Layer-wise Adaptive Rate Scaling

Layer-wise Adaptive Rate Scaling (LARS) is an optimization technique for Large Batch Training of Convolutional Networks (LARC). DNN training is only possible with online training, i.e., one epoch happens in several batches of the dataset. In order to maximize efficiency and

[1] Also available in TPUs with 60% compute speedup [1].

[2] NVIDIA reports tensor cores with 8× compute speed up on Volta/Turing architectures, 16× compute speedup on Ampère architecture and 48× average compute speed up on the Hopper architecture.

use the most resources available, the batch size is required to be increased, but this is not trivial and is met with high instability in training [126].

This instability is due to a "generalization gap", a side-effect of elevated batch size during training: for a given dataset of size N, the higher the batch size B, the fewer the number of weight updates N/B, which can prevent the model convergence. As demonstrated by Krizhevsky [69], this effect can be reduced by increasing the learning rate (LR), but the model can diverge in the initial steps if the LR gets too high.

LARS improves the convergence of neural networks on large batch training by using a different LR for each layer, where the update is defined by the ratio between weights and gradient magnitudes.

Therefore, the weight update Δw_l on the weights w_l is defined by

$$\Delta w_l = \gamma \lambda_l \nabla L(w_t), \tag{2.13}$$

where γ is a global LR and λ_l is the local LR for each layer l, calculated as

$$\lambda_l = \eta \frac{||w_l||}{||\nabla L(w_l)|| + \beta ||w_l||}, \tag{2.14}$$

where β is the weight decay.

In practical terms, You et al. [126] showed that LARS enables training ResNet-50 with batch sizes up to 32,000 without loss in accuracy. Thus, LARS is crucial if training in a multi-node or multi-GPU environment, where the effective batch size (sum of a batch in each GPU) gets large quickly.

2.3.3 Activation Checkpointing

Activation checkpointing (or gradient checkpointing) is a technique to reduce memory usage by storing only a subset of the network activations and recomputing them during a backward pass. DNN training is order of magnitudes more memory-intensive than inference, simply because it stores much more than just the network itself [23]. There are three main sources of GPU memory consumption during training:

1. Model memory: used to store weights and biases;
2. Optimizer memory: used to store gradients and momentum buffers;
3. Activation memory: the outputs of each layer consist of the forward activation memory, and the gradients computed from them during backpropagation are the backward activation memory. The sum of these consists of the activation memory.

The activation memory is often considerably larger than model and optimizer memory, as shown in Figure 2.9, which is an effect of their dependency on the model and batch sizes. In this sense, the activation checkpointing technique saves memory by discarding most of the forward activation memory, which is later recomputed during backpropagation when needed. The effect is numerically equivalent whether checkpointing is used or not, which excludes any memory-accuracy trade-off. On the other hand, there is a memory-compute trade-off given that

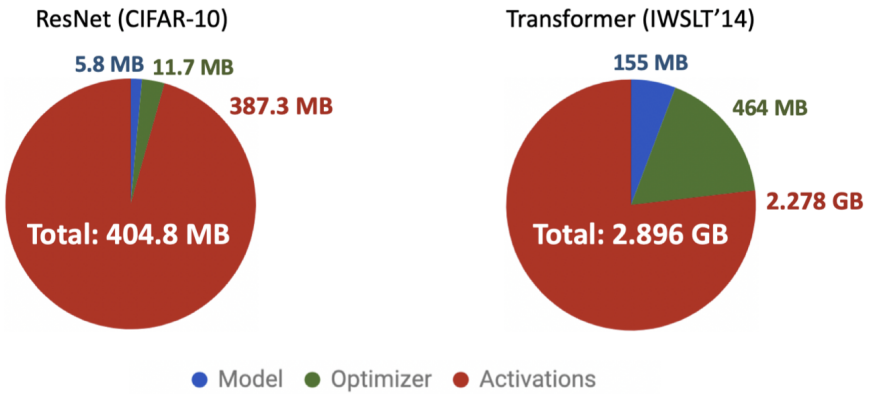


Figure 2.9: Distribution of training memory consumption for WideResNet on CIFAR-10 (left) and DC-Transformer on IWSLT'14 German to English (right). In both cases, activations represent more than 75% of everything the GPU stores while training. Figure extracted from Sharad Sohoni et al. [111].

some operations will have to be repeated, but this method can reduce nearly 50% of activation memory while inducing less than 1% increase in FLOPs (Figure 2.10).

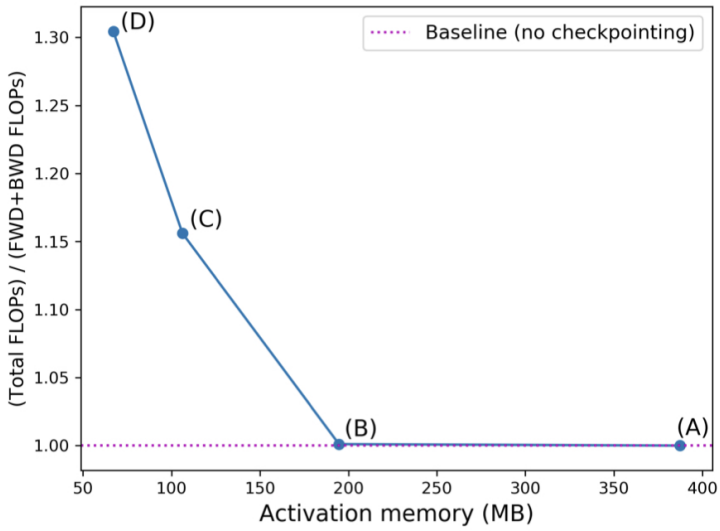


Figure 2.10: Compute FLOPs comparison for various checkpoiting methods. On the plot, (A) is the baseline of Checkpoint-None, (B) InPlace-ABN, (C) CHECKPOINT-RESIDUAL-1*, (D) CHECKPOINT-RESIDUAL-2*. Figure extracted from Sharad Sohoni et al. [111].

There are many available activation checkpointing techniques, each one with different compute-memory trade-offs. The default strategy in most DNN frameworks is known as Checkpoint-None, which simply stores all activations during training [111]. In-Place Activated Batch Normalization (InPlace-ABN) stores all activations except the outputs of normalizations and its subsequent ReLU operations, as these are computationally inexpensive to recompute but consume excessive memory [14]. Finally, there is CHECKPOINT-RESIDUAL-m*, an extension to the previous one but with the addition of a recursive approach to store only the output of every m residual block and always discard the rest [23].

Overall, the memory savings in activation checkpointing supersede the extra computing, especially considering that with extra memory available, larger networks can be used and/or the batch size can be increased. Such factor enables faster model convergence or even better final performance, which is generally desired in deep learning [23, 111].

2.3.4 Synchronized Batch Normalization

Batch Normalization (BN) is a neural network layer introduced by Ioffe and Szegedy [57] to provide faster convergence while making training more robust to weight initialization and higher learning rates. In simple terms, this layer produces y by normalizing the output x of the previous layer and keeps two trainable parameters for scale γ and shift β.

$$y = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta, \tag{2.15}$$

where μ stands for the minibatch mean, σ stands for the minibatch standard deviation, and ε is a small constant to prevent divergence or zero division errors.

However, the standard implementations of BN in public frameworks (such as Caffe, MXNet, Torch, TensorFlow, and PyTorch) normalize the data within each GPU, which means it will not use the actual training batch size. In computer vision tasks using small batch sizes, this can lead to intensive performance degradation [130], and the simplest solution is to replace the layer for synchronized batch normalization (SyncBN).

SyncBN computes the global values of $\sum x$ and $\sum x^2$ to then calculate a synchronized μ and σ. Similarly, there is a second synchronization for the gradients during backpropagation. Finally, global γ and β are updated.

Importantly, the synchronization does slow down the training depending on the number of SyncBN layers, but it does not become a significant time-consuming step. In the end, SyncBN generally increases model quality on distributed training scenarios [130].

2.4 Bayesian Estimation Technique

Statistical inference usually requires comparing two or more groups to understand which is larger than the other or simply to verify if they are different.

A statistical hypothesis test is often employed in such comparisons, but this approach is seldom correctly conducted, and the results are easily misinterpreted. The choice of the statistical test, null hypothesis, significance threshold, and more are usually inherited by traditions [63]. This leads to arbitrary checks that would not formally satisfy the requirements and tend to incorrectly go against the null hypothesis [41].

Thereafter, comparing different samples based on estimation techniques is far more effective. By such methods, we intend to measure the difference among said groups and estimate the associated uncertainty instead of performing a simple binary check.

A strong statistical model is required because true differences are noise-related, preventing conclusions from being drawn from differences calculated directly from the observed data. For that end, we employ the BEST technique, which is a complete replacement for the t-test by embracing uncertainty while maintaining simplicity [116].

Bayesian Estimation (BEST) models the data as samples from a t-distribution, which is a Gaussian-like distribution with the addition of a normality parameter ν to control the prevalence of outliers [71]. Thus, given two samples y_1 and y_2, we take the following prior distributions in the model

$$\begin{aligned}
\mu_1 &\sim \text{Normal}(\hat{\mu}, 2\,\hat{\sigma}) \\
\mu_2 &\sim \text{Normal}(\hat{\mu}, 2\,\hat{\sigma}) \\
\log(\sigma_1 \,/\, \hat{\sigma}) &\sim \text{Uniform}(\log(1\,/\,2), \log(2)) \\
\log(\sigma_2 \,/\, \hat{\sigma}) &\sim \text{Uniform}(\log(1\,/\,2), \log(2)) \\
\nu &\sim \text{Exponential}(1\,/\,27.5) + 2.5 \\
y_1 &\sim t_\nu(\mu_1, \sigma_1) \\
y_2 &\sim t_\nu(\mu_2, \sigma_2),
\end{aligned}$$

where $\hat{\mu}$ is the pooled mean and $\hat{\sigma}$ is the pooled standard deviation [119], so the goal is to check if we are indeed able to distinguish two distributions from the pooled statistics.

This version of the model differs from the one originally presented by Kruschke [71] as the goal of BEST in this Master's thesis is to compare different models by the balanced accuracy. First, we rescale the standard deviation choice of priors, as the accuracy cannot go beyond the range of $(0, 1)$ [116]. Moreover, the normality lower bound is also increased to 2.5, so that most of the distribution is kept within $\mu \pm 5\sigma$ while preventing strong outliers from generating exploding standard deviations [119].

2.5 Conclusion

All the methods mentioned above are used as building blocks of our experiments. Each contrastive learning technique (SimCLR, SwAV, BarlowTwins, and SupCon) will be tried as part of the scene classification training process. Given how expensive these methods are, however, we implement the optimizations throughout the process in order to improve training speed. Finally, we use bayesian estimation to discover which technique had the best performance and to measure the confidence that it is indeed the best overall.

Chapter 3

Related Work

Child sexual abuse material (CSAM) has been propagating exponentially on the online media in the last couple of decades, but it is still difficult for law enforcement to efficiently recognize CSAM and identify the victims. Scene understanding and recognition can thus be leveraged to aid in this task and rapidly reduce the millions of images awaiting processing without even needing for sensitive material during training.

In this chapter, we introduce the topic of scene recognition and highlight the challenges that make it particularly complex compared to object recognition (Section 3.1). We overview the research in this field, emphasizing the SSL approaches. Also, we summarize recent methods for CSAM recognition, focusing on machine learning-based ones and how scene recognition can help this problem (Section 3.2).

3.1 Scene Recognition

Human beings are profoundly efficient in analyzing their environment and important events, but machines and algorithms have long stayed behind in this topic. Even humans have problems fully comprehending certain scenes, but we are fairly good at classifying them, and when it is not possible to understand them globally, we find ways to grasp them from local parts. As we can check in Figure 3.1, sometimes we take time to find Waldo or to understand what the people in the image are doing, but we know almost instantly that the scene is an amusement park.

In that sense, the task of scene recognition has the sole purpose of distinguishing scenes: semantically coherent views of real-world environments containing objects, textures, and a background spatially separated [35, 51]. This is a well-studied field with a particular interest in surveillance, autonomous driving, and robotics navigation, but still lacking in performance, especially within indoor classification [100].

On the small datasets, “simple” DNN models such as AlexNet trained on ImageNet achieve accuracy comparable with the best non-deep learning methods [70], but some recent approaches have finally reached valuable performance for real-world applications [129]. In this section, we explain the problems involved in scene classification and how they differ from other deep learning tasks and also show the most recent advances being done in the field, with special attention to self-supervised methods.

Figure 3.1: Despite a large number of people, objects, and actions, we can readily verify that the scene is an amusement park. Figure extracted from the book “Where is Waldo?”, by Martin Handford.

3.1.1 Challenges

Scene classification is closely related to object and texture classification, but with the presence of multiple objects, textures, backgrounds, and intricate relations between them. Thus, one could imagine scene classification as multiple local classifications happening simultaneously to provide a context and a set of items or actions. However, such a process would be practically impossible: the high variance and natural complexity of real-world scenes are too large and, in fact, simpler visual cues should be used for classification [129].

Humans can quickly and unconsciously spot small signs in the background and within object disposition to classify a scene [35], but the dense amount of information united with pixel-level cues is not easily grasped by machine models. Overall, this is represented by large **intraclass variation** and **interclass semantic ambiguity**.

First, the **intraclass variation** initially arises from the wide range of objects, backgrounds, and human actions a scene can display, which is shown in Figure 3.2. Beyond that, one single scene accepts large variations in object scale, illumination, occlusion, clutter, shading, blur, contrast, motion, or can be shot from multiple viewpoints and with different item distributions [129].

Next, the **interclass semantic ambiguity** is related to the intersections of objects, textures, and backgrounds among different classes, with subtle changes in the numerical or visual distri-

Figure 3.2: Depiction of intraclass variation in a dataset. All images have the classroom label and were taken from Places365 [134]. It is possible to see variations in objects, presence of people, viewpoints, background (open vs. walls), colors, lighting, and overall content.

butions to define the scene [84]. As shown in Figure 3.3, this is even a challenge for humans, as scene classification is sometimes subjective and prone to cultural differences, where two very distinct places for some can be taken as the same scene to others. This difficulty becomes more noticeable the higher the total number of categories a dataset has, with a single scene actually belonging to multiple classes [12].

(a) Bedroom (b) Children Room (c) Dorm Room (d) Hospital Room (e) Hotel Room

Figure 3.3: Illustration of interclass semantic ambiguity. All images below share objects, general layout and distribution of items, but small cues put them in their own different classes. Images taken from Places365 [134].

Additionally, the human brain has specialized areas to represent the geometric structure of scenes from background elements [35], which is a topic where neural networks still face great limitations unless specific handcrafted hints and constraints are given initially to the system [58, 115].

Finally, similar issues exist in other computer vision and machine learning problems, from where several specialized solutions got inspiration.

3.1.2 Deep Learning and Scene Classification

For a long time, scene classification relied on pure handcrafted solutions, but the current most successful solutions in scene classification use deep learning and scene representation. Here we provide a simple review of the work done in the field and its current state-of-the-art.

Neural Networks used for scene classification are typically convolutional nets (CNN) with popular architectures such as VGGNet or ResNet. However, CNNs are not usable in this field without being trained on large amounts of data, meaning most of the works perform transfer learning [129].

However, transfer learning is usually limited by the similarity between the source and target domains, meaning that ImageNet pretrained models are, on average, less effective than Places pretrained versions. This happens because ImageNet is considered an object-centric dataset, producing models with object descriptors, i.e., the extracted features are for individual or homogeneous objects in specific scales. For example, the class Ball in ImageNet would hardly provide images where the ball would be mostly hidden or occupying less than a third of the image, so the model learns that balls are only meaningful in certain specific conditions [132].

On the other hand, it is far more common to find ImageNet pretrained CNNs than Places pretrained ones, forcing some researchers to use the former models. After a pretrained model is selected, several works concentrate on developing specialized finetuning routines which can extract the most scene-centric features in the target dataset [134].

These models are designed to extract effective scene representations by changing the usual CNN architecture. Simple finetuning can even harm performance, as the most common target datasets are too small compared with ImageNet and Places [134]. In this case, data augmentation can sometimes help, but changes in scale worsen results given that it generates completely different images with the same label, leading the model to respond equally to most images, a consequence of the interclass ambiguity [53].

Initially, most works designed CNN architectures that still extract global features, i.e., predict the scene classes from the whole image. This approach is faster overall since it does not go too far from the original deep learning pipeline, but the performance is highly dependent on the richness of the scene datasets. For example, GAP-CNN replaces fully connected layers for global average pooling (GAP) to focus on class-specific regions while using fewer parameters [133]; Hierarchical LSTM (HLSTM) uses four LSTM modules in an attempt to capture spatial dependencies [136]; DL-CNN proposes Dictionary Learning (DL) layers to obtain sparse representations and reduce the total number of a parameter; finally, the Spatially Unstructured (SU) layer was designed to help cope with layout deformations and scale changes in scene recognition [47].

However, global CNN feature methods do not exploit a scene's visual or semantic relations. In this sense, some methods propose training from patches of the original image to derive robust representations against geometrical variations. These image patches can be produced randomly or extracted using a different algorithm. Later, encoding techniques that produce spatially invariant representations are applied to produce spatial invariant representations: FV-CNN [29] and MFAFVNet [78] use Improved Fisher Vector (IFV) while MOP-CNN [40] and SDO [28] employ a Vector of Locally Aggregated Descriptors (VLAD) to cluster local features. In the end, such an approach heavily increases the number of images to be processed and the overall

training complexity, so it tends to be orders of magnitude more expensive than common deep learning while introducing excessive noise, as most patches do not help classify the scene [129].

Additionally, object detectors have been used to help with identification of important regions within the scene [80], such as Fast-RCNN [39], SSD [82] and YOLO [104, 105]. Using the patch generator from the previous approach in this pipeline is also possible to use semantic features from different scales. With two semantic features, one for larger and another for small objects, it is already possible to represent the whole scene [34]. This forces the model to pay attention to particular objects and helps deal with interclass semantic ambiguity.

In other words, as shown in Figure 3.4, to distinguish between two rooms, the model must consider many parts of the scene to understand the context. Additionally, Qiu et al. [99] showed that a scene is constituted of minor and essential objects, and only the essential ones matter for the final scene classification, while minor elements can be removed without impact. However, the number of minor objects is usually much higher and is basically noise, which can be accounted as factual evidence of the intraclass variation in scene recognition.

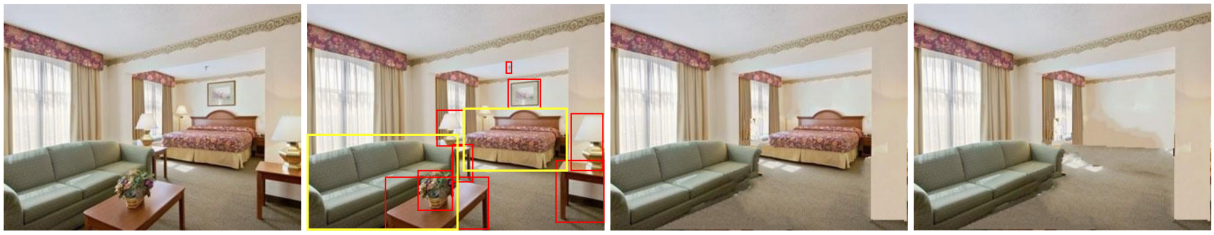


Figure 3.4: A hotel room image is an input (a), and the essential and minor objects are detected (b). The image with essential elements preserved but minor ones inpainted (c) is still classified as a hotel room. However, in this case, the bed is inpainted, an essential object in a hotel room. The scene will be recognized as a living room instead. Figure extracted from Qiu et al. [99].

Semantic feature methods have demonstrated good performance [22, 72, 131], but rely on the quality of object detectors. Typically, this dependency introduces two-stage training [123], meaning that learning has to be split between the object detector and classifier.

In this context, several scene classification methods have used multiple layers' outputs to improve scene representation. The high CNN layers are too compact for the dense level features required for scene recognition, keeping only large objects, while the low CNN layers still keep the small ones [121]. Thus, harnessing multi-layer features reaches overall better representations, but careful feature fusion design is required [81]. Ultimately, this is the most structurally complicated approach and the fused features generate high-dimensional arrays, making the model overfit easily the more layers are used [124].

In conclusion, most deep learning works so far have relied on pretrained neural networks and small-scale datasets to solve the problem of scene classification. The development of specific neural network architectures to work this task was, however, crucial, and the current state-of-the-art FTOTLM [81] shows top-1 accuracy of 94.1% on MIT Indoors [100] and 85.2% on SUN RGB-D [114] with a multi-layer architecture. In large-scale datasets, the Layout Graph Network (LGN) [22] is the state-of-the-art supervised model, having reached 56.5% top-1 accuracy on Places365-Standard.

3.1.3 Self-Supervised Learning applied to Scene Recognition

Scene recognition has made significant progress in the last couple of decades, but the challenging nature of this task maintains some unsolved problems. As mentioned before, there is a high dependence on dataset size and richness, and while much of the success of deep learning depends on labeled data, it is always very limited to rely on that.

In this scenario, self-supervised learning becomes very appealing when looking to increase the model's richness and diversity without increasing the cost, as there is no human labeling involved. Furthermore, Quattoni and Torralba [100] pointed out that scene classification "is related to work on learning distance functions for visual recognition", also known as *metric learning*. Such a remark was made a decade before contrastive learning became a topic in the deep learning field, yet it shall serve as the initial motivation for applying contrastive methods to indoor classification.

However, to the best of our knowledge, few works try to solve scene classification with self-supervised learning. In scene semantic segmentation, McCormac et al. [88] showed an improved model on NYUv2 and SUN RGB-D benchmarks using a model pretrained on a 5M image 3D rendered dataset called SceneNet RGB-D. In order to tackle the PASCAL VOC benchmark (11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations), Ren and Lee [106] proposed a multi-task learning approach that pretrains an AlexNet model to predict edges, surface normals, and depth maps from SceneNet RGB-D images and a discriminator to distinguish the synthetic images from the Places365 images. Similarly, She and Xu [112] added a SimCLR contrastive loss as a new pretext task, setting as positive pairs all viewpoints of a single scene. The aforementioned methods were able to reach comparable performance or even outperform other SSL techniques using the same neural network architecture, but underperformed when compared to the state-of-the-art method for the tested benchmarks.

Whereas scene classification with SSL has not been found in the literature by this thesis' researchers, the articles which propose new self-supervised techniques commonly check the performance of their proposed models on Places205 (2,5 million images), an older version of the Places365 dataset. In that sense, self-supervised methods present the state-of-the-art with SwAV having reached 56.7% top-1 accuracy [17] using a ResNet-50, while SEER set the mark of 69% with a RegNetY-128GF-10B-384px [42]. Hence other contrastive learning and self-supervised tools should be most suited to perform recognition in indoor scenes. It is noticeable that these values of accuracy are much less than what is usually seen on other image classification datasets, such as ImageNet, and this is due to the greater number of high-level features scene recognition demands [42] and the difficulties involved in dealing with intraclass variation and interclass ambiguity.

In short, scene recognition demands richer datasets but, simultaneously, supervised learning is not scalable as it explicitly requires labeled data. Therefore, methods looking to reduce the dependence on labeled scene images seem to be good opportunities to improve deep learning in scene classification.

3.2 CSAM Recognition

Easy access and impunity led to a frenzied growth in the consumption and distribution of CSAM over the past few years, with hundreds of millions of images circulating through the web today [15]. In a world where access to CSAM is oversimplified and bears a low risk of getting caught, people may stumble upon such content and develop more interest, while others also make it an income source [89]. In this section, we list some recent approaches to CSAM recognition and several other tools used to help.

The automation of CSAM recognition can protect adults and children, more easily blocking that sort of content in social media and thus preventing trauma [76], but this is still a challenging field of research. Such datasets are and must stay accessible exclusively to law enforcement personnel, dramatically increasing the difficulties in comparing possible models or performing benchmark evaluations [73].

Moreover, NCMEC shows that hash comparison tools used by Internet Service Providers (ISPs) are already the largest source for reports today. However, when dealing with never-before-seen data, law enforcement personnel are the exclusive mechanism responsible for auditing CSAM, which not only is the source of bias but also of psychological strain on the agents' mental health [85].

In light of these issues, researchers have attempted to design more scalable and reliable methodologies to tackle CSAM recognition. Image hash databases, web crawlers, and file metadata are most commonly used to detect CSAM sources and find the criminals [46, 95, 113]. In a more versatile method, NuDetective de Castro Polastro and da Silva Eleuterio [31] and iCOP Peersman et al. [97] make use of handcrafted nudity image descriptors to build fast CSAM detectors, and Sae-Bae et al. [110] used textures and facial distances to distinguish adult from child nudity. However, these solutions are sensitive to slight modifications, demanding constant updates to stay functional.

With the recent developments in computer vision detectors, machine learning algorithms started to be used for CSAM recognition as well. Ground-up training and transfer learning of CNNs were shown to outperform current forensic and commercial tools easily and can be made into portable tools for search and seizure procedures [120]. Moreover, Gangwar et al. [37] demonstrated that pornography and age-group recognition could be leveraged for CSAM with substantial gains in accuracy for binary classification, and Macedo et al. [85] used a single-model estimation of child presence, age, and gender to improve the performance. Rondeau [109] arguments apparent age and nudity detection could also be leveraged for CSAM, while Castrillón-Santana et al. [19] focused on non-adult people detection in images for CSAM age estimation, reaching over 90% accuracy on their proposed dataset.

All these studies firmly outline that deep learning can contribute to child abuse inquiries. However, most published works focus on detecting children and nudity/pornography [5, 59, 98] and highlight the difficulties of finding large annotated CSAM databases to experiment with, even with the help of law enforcement agents. Several works focus exclusively on age estimation within CSAM, and children's images are collected to build datasets that help accomplish such tasks. For instance, Anda et al. [4] proposed VisAGe and Castrillón-Santana et al. [19] AgeMega, databases focused on age estimation with thousands of underage and adult facial images, while Castrillón-Santana et al. [19], Chaves et al. [20], Gangwar et al. [37] gather images

from several age estimation databases. Beyond that, Al-Nabki et al. [3], Tabone et al. [117] included sexual organ detection to aid CSAM detection, while Yiallourou et al. [125] used facial expressions to distinguish sensitive from non-sensitive material, which is one of the main challenges in the field. Lee et al. [76] call attention to the fact that "the best results can be achieved if multiple methods are used in combination", meaning that CSAM is a complex enough problem that age, nudity, and pornography detectors do not fulfill all the roles for possible feature extractors that can aid on this task.

On this matter, Laranjeira et al. [73] designed an aggregated analysis pipeline to aid CSAM recognition research by showing plots of statistical distributions of some features, such as scene, objects, faces, skin tone, nudity, image sharpness and more, from the region-based annotated child pornography dataset (RCPD) [85], without publicizing information on individual samples. This dataset is a private database used internally by the Brazilian Federal Police and contains 2138 images among CSAM, pornography and non-sensitive categories. There is, however, another image set with over 45 thousand images available, which was used by this thesis to conduct experiments.

In this context, most databases used for CSAM recognition could suffer from biased distributions, not actually representing what is found in the real world. Laranjeira et al. [73] show that CSA data shared online differs from reported cases of physical abuse in Brazil in terms of race: "most reported victims of child sexual abuse in Brazil are black and brown girls from 8 to 14 years old, but RCPD depicts mostly white children being abused and most of the material apprehended in Brazil has different tendencies than reports of sexual abuse with physical contact."

With this in mind, CSAM classification could take advantage of non-direct CSAM recognition, and scene recognition should greatly help. Bursztein et al. [15] emphasize the investigation should optimally consider scene information as means to cluster similar imagery and "report objects that are present, its environment, and potentially identify landmarks that will help locate the region where an abusive image originated". Experts use scene information to gather context and understand what is happening in the image or video, especially in low-resolution scenarios and when people are covered or with their backs turned [67]. CSAM distribution is a massive problem worldwide, and, for the sake of the children under abuse, combating it demands new data-driven scalable solutions that can find patterns human beings would not often notice.

Chapter 4

Methodology

This work focuses on quantitative and reproducible research, thus simplifying possible replication or extension endeavors. We follow a methodology split into self-supervised pretraining and cross-validation model finetuning. Our goal is to verify that it is possible to help CSAM investigation with indoor recognition.

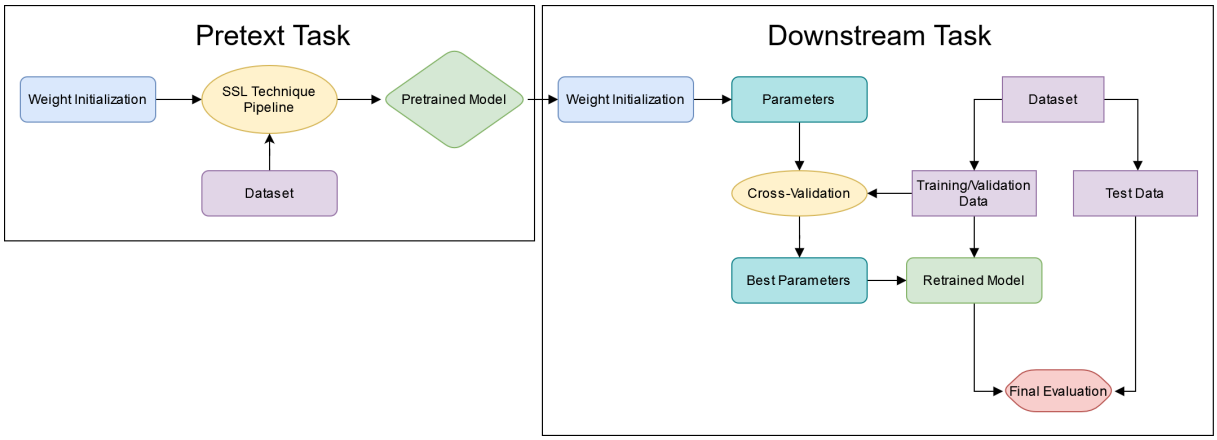


Figure 4.1: Full pipeline flowchart defining the two different set of experiments in SSL. The pretext task uses unlabeled data and runs the SSL technique, while the downstream task uses labeled data from our target task and initializes from SSL pretrained models.

We aim to investigate the reliability of self-supervision for scene recognition and if large amounts of synthetic data can be employed to easily replace the requirement for real indoor environments. We hypothesize that photorealistic synthetic data can bring more diversity to the final representation, as it contains images of the same environment in different lighting and angles, providing better-detailed features with less semantic ambiguity than real datasets.

With such intent, we initially selected real and synthetic datasets suitable for the task, filtered in indoors-related categories, and used them to finetune pretrained models. ImageNet pretrained models come from public "model zoos" while scene-centric SSL pretrained models are pretrained by ourselves. We built a small test dataset that could lighten some of the model flaws and evaluated the best model against real CSAM to determine if we are indeed able to help with this material.

The full training and testing pipeline is presented in Figure 4.1. In the following sections, we explain each of the sub-pipelines for pretraining (Section 4.1) and finetuning (Section 4.2).

4.1 Self-Supervised Learning Pipeline

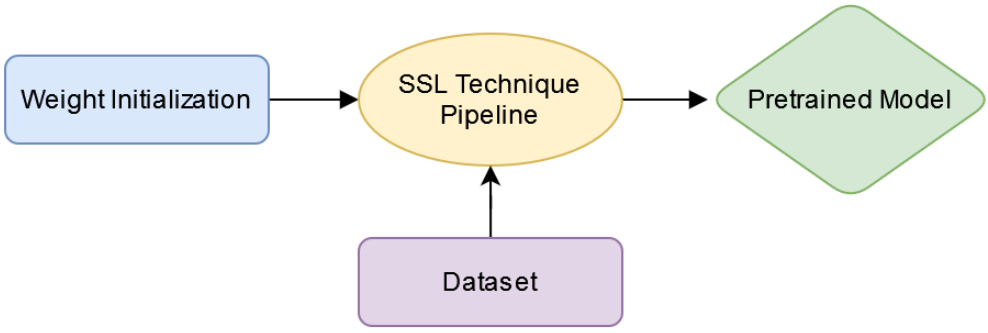


Figure 4.2: Self-supervised flowchart used for model pretraining. A model is initialized and trained following a specific SSL procedure on the selected dataset. This process outputs a pretrained model, which will be further specialized on downstream tasks.

1. **Dataset**: the dataset used for model pretraining, e.g., Hypersim [107]. For self-supervised learning, labels are not required, so labeled and unlabeled data can be used.
2. **Weight Initialization**: the network weights can start randomly or come from another pretrained model, e.g., an ImageNet model available on a public model zoo.
3. **SSL Technique Pipeline**: the pretraining step in which a specific self-supervised pipeline is used, e.g., SimCLR [24], SwAV [17].
4. **Pretrained Model**: the final pretrained model weights are used in weight initialization on downstream tasks in the finetuning step.

Self-supervision is a tool for helping generate good representations from unlabeled data. This deep learning approach has proved to be more flexible and reliable than traditional supervised learning on several applications [36, 62]. With this line of thought, we use SSL to increase the general performance of machine learning models for scene classification.

In particular, SSL starts with a pretext task (Figure 4.1), which is a pretraining task in an unlabeled dataset before the target dataset is used. This step is developed from the selected datasets to understand if we can aid scene recognition with large scene datasets.

With the dataset in hand, the pretext task pipeline is always a contrastive learning method, following one of the techniques listed in Section 2.2, and a general schema of this process is shown in Figure 4.2. As explained earlier, SSL does not benefit from a validation set, so it is not reliable to early stop the training when the validation loss is at its minimum. Therefore, we train the model for all predefined epochs, which makes it one of the most expensive steps of this Masters's thesis.

Moreover, the proposed pipeline allows for a flexible generation of pretrained models. This is a result of the number of SSL techniques used and the fact that they all have ImageNet SSL models available. Therefore, we produce two models for a single pretraining dataset and SSL technique: one trained from scratch (random weight initialization) and another trained from the ImageNet models' weights.

Finally, considering the adoption of SwAV, SimCLR, BarlowTwins, and Supervised Contrastive techniques, for a single pretraining dataset, we produce eight pretrained models. We decided not to combine different techniques to reduce the total number of experiments and computation costs.

4.2 Cross-Validation Model Finetuning Pipeline

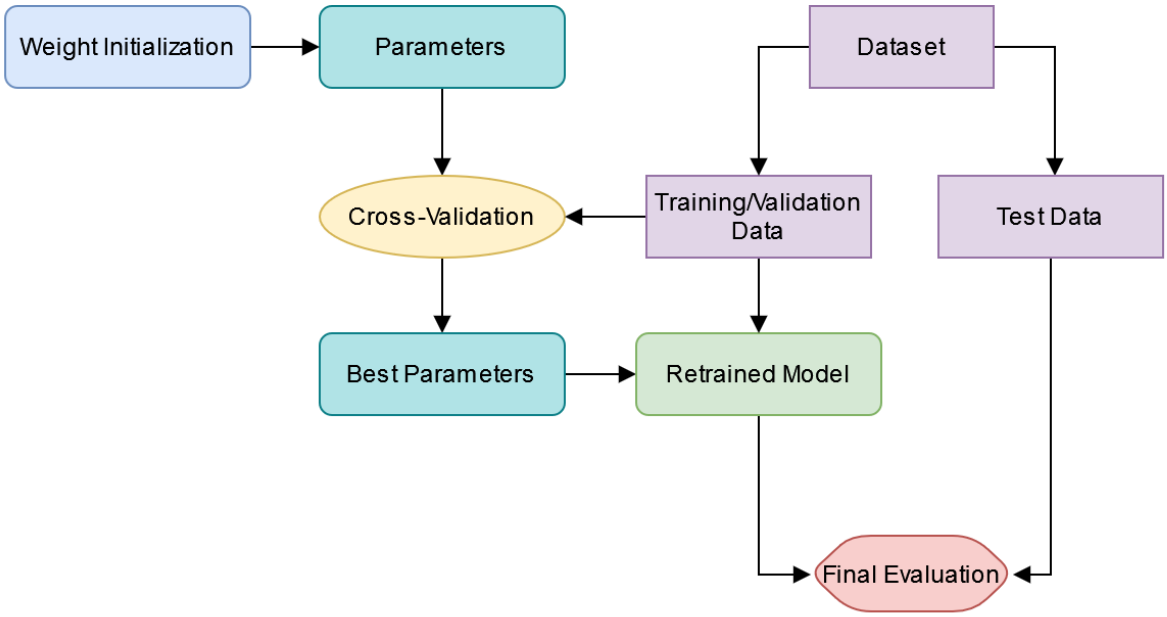


Figure 4.3: Cross-validation flowchart used for model finetuning. The dataset is split into training, validation, and test sets, and the search for the best model happens using a k-fold cross-validation protocol. Given that the best hyperparameters are found, the best model is then trained on the whole training set and evaluated on the test set.

1. **Weight Initialization**: for finetuning, the network weights come from a pretrained model: either an ImageNet model available on a public model zoo or one of the models produced on the last SSL pretext task.

2. **Parameters**: grid search of hyperparameters. Each combination of hyperparameters define a cross-validation experiment. These include, but are not limited to learning rate base value, self-supervised technique, and number of pretrained epochs.

3. **Dataset**: the dataset defines the downstream task being tackled. All datasets in the model finetuning step were human labeled, e.g., Places365.

4. **Training/Validation Data**: the dataset is split into training, validation and test data in a stratified manner. The training and validation data will be used in the cross-validation to find the best possible model.

5. **Test Data**: the dataset is split into training, validation and test data in a stratified manner. The test data is held off for the final evaluation, when the best model has been selected.

6. **Cross-validation**: k-fold cross-validation in the training data in a stratified manner. A validation subset is also used on this step for early stopping.

7. **Best Parameters**: cross-validation produces k values for each metric, which can be used to select a combination of the best parameters. A box-plot, mean plus standard deviation or student t-test are viable selection techniques.

8. **Retrained Model**: the best parameters and all training data are used to produced a re-trained model, the best found model.

9. **Final Evaluation**: the best-trained model is then evaluated on the test set, which was kept away until this part. We report metrics and scores of this final evaluation.

In this work, all downstream tasks are classification problems. Therefore, we follow the k-fold cross-validation schema shown in Figure 4.3 to find the best possible model for a given dataset.

Consequentially, k-fold cross-validation considerably increases the total number of experiments, as each experiment becomes a loop of k new iterations. This problem influenced the design of our experiments and led to constraints on what we were able to do. In turn, such a procedure makes our results more reliable and statistically more accurate.

As will be shown further, the validation set comes from the original dataset. It is relatively small compared to the size of our training set, which initially led to a heavy similarity of metrics, making it statistically unreliable to select the best possible model.

Thus, cross-validation increases our results' trustworthiness and disambiguate each model's quality more clearly.

Chapter 5

Experimental Results

In this chapter, we start showing the datasets selected for the tasks and explaining how they were remapped, filtered and joined together to construct our pretext and target tasks (Section 5.1). Next, we present the experimental design following the pipelines defined in Chapter 4 to understand the best conditions for scene recognition in each scenario (Section 5.2). Then, we execute the experiments to find the best model and, after it is selected, evaluate it on images outside of the target set and also hand it to the authorities in possession of real CSAM data for a final test on this kind of material (Section 5.3). Finally, we conclude with some comments on the hardware used and implementation details (Section 5.4).

5.1 Datasets

In the context of scene classification, we chose scene-centric datasets with a high diversity of displayed objects and features.

In this section, we present scene datasets from the literature (Section 5.1.1) and the ones we built for our indoor classification task (Section 5.1.2). These datasets can be realistic or synthetic; we subset and reorganized them for the best usability towards our objectives. Also, we introduce our Litmus Test dataset, a "litmus test" to check if the model performance generalizes (Section 5.1.3). Table 5.1 summarizes all datasets used for pretext and downstream tasks.

Table 5.1: Datasets we used in scene recognition for pretext and downstream tasks. Size stands for the total number of images, Density stands for the range of images for each class, and Synth stands for synthetic datasets. For the synthetic datasets, a Class is defined as a single frame, i.e., the photorealistic render and its different visualization maps. †Custom datasets.

Dataset	Size	Density	Objects	Scenes	Classes	Synth	Year
Places365 [134]	8,496,949	3168 – 40,100	–	365	365	✗	2017
InteriorNet [77]	4,000,000	10	50	9	40,000	✓	2018
Hypersim [107]	1,432,480	9 – 18	20	12	143,248	✓	2021
OpenRooms [79]	118,233	3	44	1287	39,411	✓	2021
Places8 (ours)†	407,640	14,137 – 111,724	–	23	8	✗	2022
Indoors.all (ours)†	7,778,152	3 – 40,100	> 114	1467	222,818	✓	2022
Indoors.real (ours)†	2,490,632	3168 – 40,100	–	159	159	✗	2022
Litmus Test (ours)	80	10	–	8	8	✗	2022

5.1.1 Literature Datasets

Places365 is a 10 million image dataset focused on scene recognition. It is composed of four different subsets, Places88, Places205, Places365-Standard, and Places365-Challenge, in which the Places365 family is the last updated version and thus the one recommended for training machine learning models. This dataset is composed of real images extracted from image search engines, their categories all come from WordNet scene semantic terms, and its ground truth labels were produced with Amazon Mechanical Turk.

WordNet provides a hierarchy graph among the classes, split into three major groups: indoors, natural outdoors, and urban outdoors (Figure 5.1). We used a subset of Places365-Challenge composed only of indoor scenes, totaling 2,490,632 images split among 159 categories and ranging between 3,168 and 40,100 images per class.

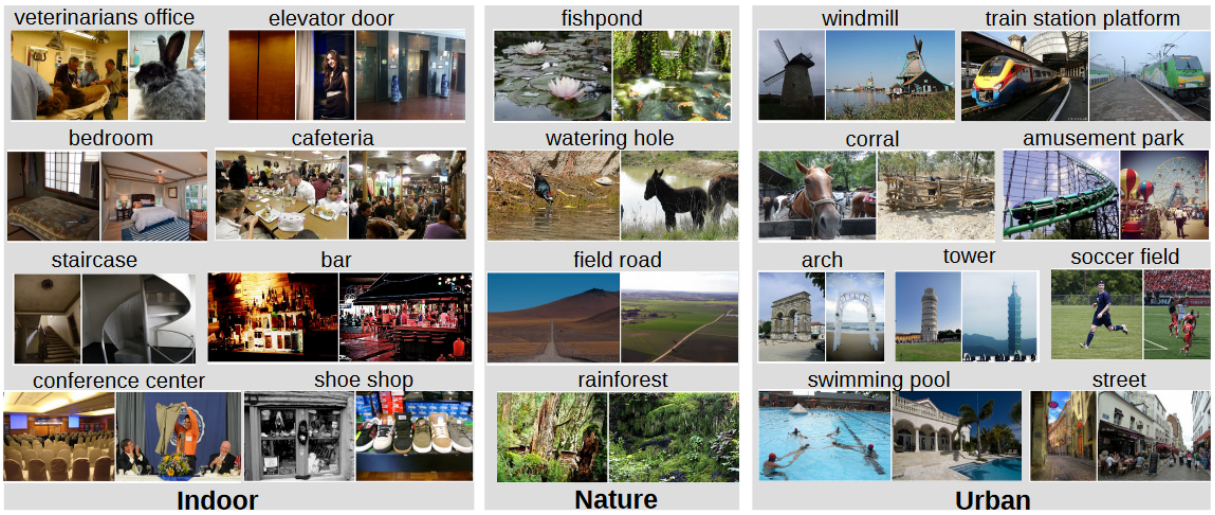


Figure 5.1: Places365 scenes are grouped into three major groups: indoors, natural outdoors, and urban outdoors. Figure extracted from Zhou et al. [134].

InteriorNet is an end-to-end pipeline able to render an RGB-D-inertial[1] benchmark for large-scale interior scene understanding. The publicly available dataset contains 4 million images of 9 environments: bedroom, guest room, bathroom, living room, kitchen, dining room, balcony, study, and kids' room, ranging from 58,600 to 1,724,400 images per environment. Beyond that, each frame has eight different views: original, diffuse albedo, surface normal world/camera, depth, illumination, and random lighting illumination/camera, which some renders can be seen in Figure 5.2.

Hypersim is a photorealistic synthetic dataset for holistic indoor scene understanding. It offers many images with per-pixel ground truth labels generated by a computer graphics engine using only publicly available 3D assets. The renders display complete scene geometry, material, lighting information, and scene segmentation for every scene while factoring every image into

[1]An RGB-D image combines an RGB image and its corresponding depth image. A depth image is an image channel in which each pixel relates to a distance between the image plane and the corresponding object in the RGB image.

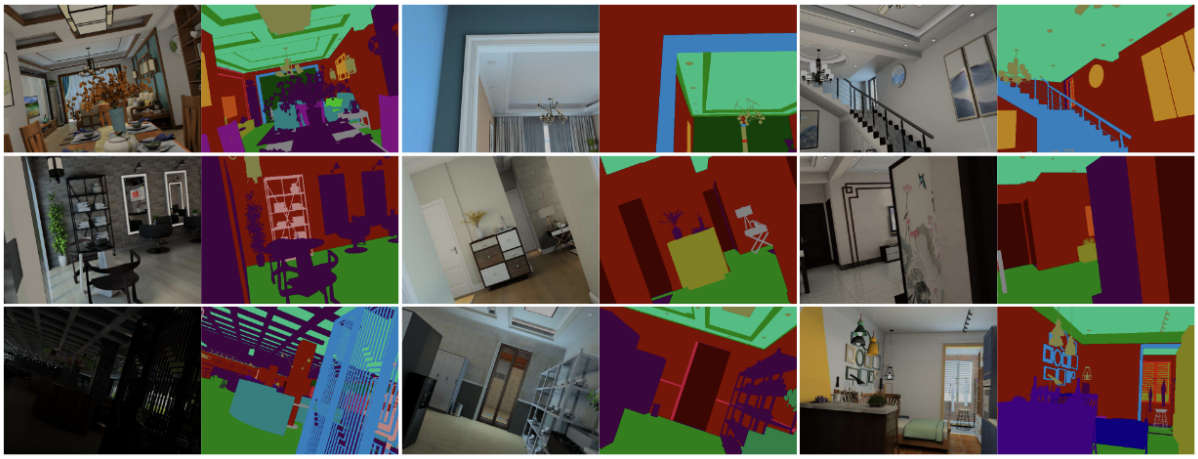


Figure 5.2: Semantic segmentation of different frames of the InteriorNet dataset. Each photorealistic render is followed by its segmentation map on the right. From left to right, top to bottom: a dining room, living room, stairs, hair saloon, hallway, lobby, office, locker room, and bedroom. Figure extracted from Li et al. [77].

diffuse reflectance, diffuse illumination, and a non-diffuse residual term that captures view-dependent lighting effects [107].

This dataset is composed of 1,432,480 images, around 10 images for each frame, with a maximum of 18 views for each frame, including color, gamma, surface normals from the camera and world matrices, texture coordinates, semantic instantiation and segmentation, depth, tone maps, diffuse illumination, diffuse reflectance, entity id, difference, residual, Lambertian and non-Lambertian maps. We show the used views in Figure 5.3, in which only one of the available surface normal views was selected, and the difference view was removed due to being mostly black images.

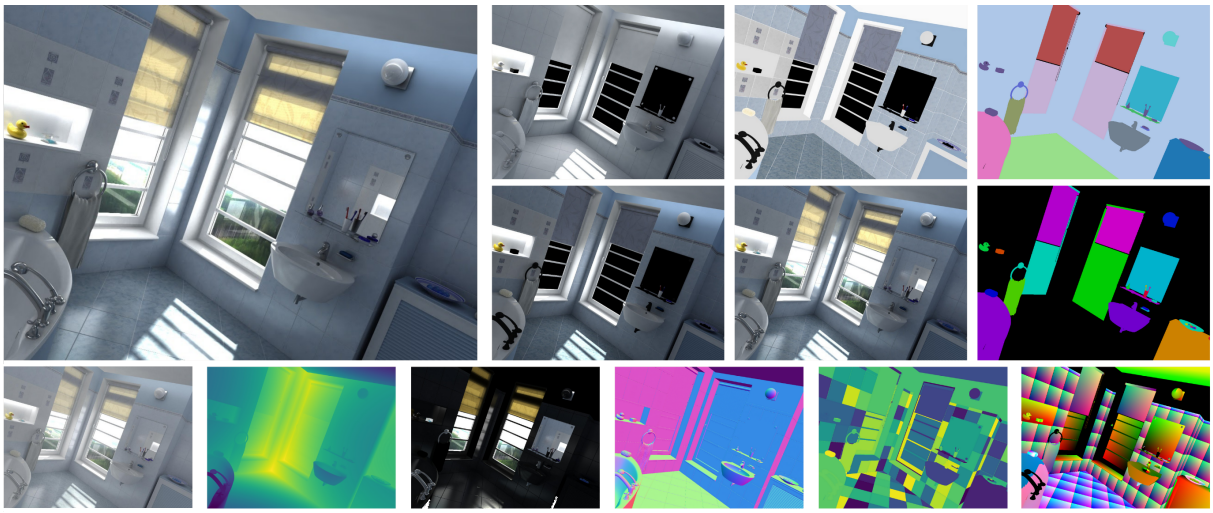


Figure 5.3: Hypersim views of the same frame. From left to right, top to bottom: color, diffuse illumination, tonemap, semantic segmentation, diffuse reflectance, Lambertian, semantic instantiation, non-Lambertian, depth, residual, surface normal, entity id, and texture maps. Figure extracted from Roberts et al. [107].

OpenRooms is a framework for photorealistic indoor scene datasets. It can produce high dynamic range (HDR) images with ground-truth depths, surface normals, spatially-varying bidirectional reflectance distribution function (BRDF), light sources, and per-pixel spatially-varying lighting and visibility masks for every light source (Figure 5.4). Considering this framework, the number of available images is not the maximum possible, and users can render more images from the given assets as they need. Despite that, we used the publicly available dataset, composed of 118,233 images, with three images per frame, resulting in 39,411 frames [79].

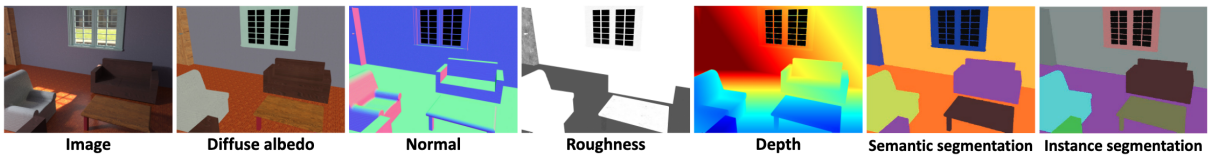


Figure 5.4: Some of the possible views from OpenRooms for the same single frame. The publicly available dataset contains the original rendered image and two versions of the diffuse albedo view. Figure extracted from Li et al. [79].

5.1.2 Custom Datasets

In our experiments, remapping and filtering the datasets mentioned in the last section were employed to build the target task for this Master's thesis.

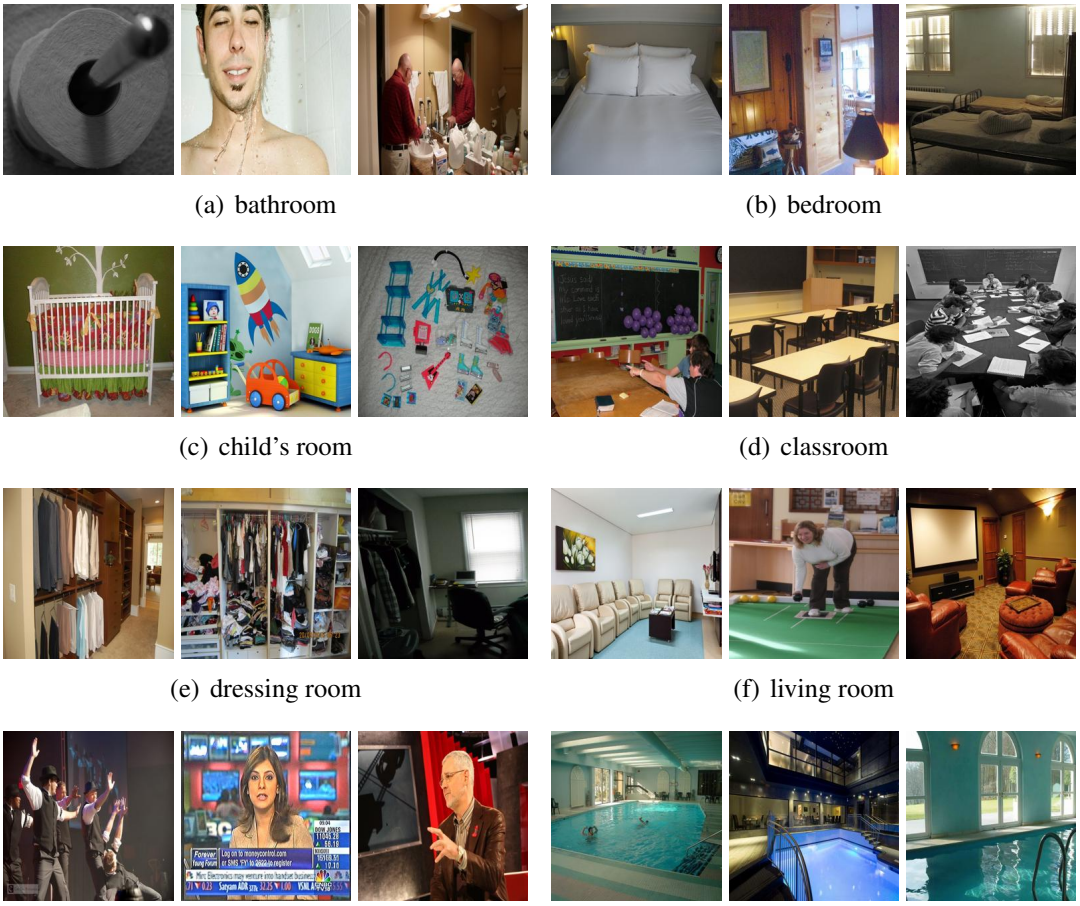
Places8

All datasets mentioned earlier are primarily used for self-supervised pretraining in indoor environments, but we constructed a smaller dataset for finetuning purposes. Places8 is a subset of Places365-Challenge focused on environments most common in CSAM.

Initially, to increase the number of images per class and therefore raise the intraclass diversity while reducing the interclass ambiguity, we grouped Places365-Challenge indoor classes from 159 to 62 new categories following WordNet synonyms and sometimes direct hyponyms or related words. For example, *bedroom* and *bedchamber* were joined, while *child room* was kept in a separate category given its importance in CSA investigation. The complete remapping can be seen in Table 5.2 under "Original Categories".

Next, we filtered the remapped dataset into 8 final classes from 23 different scenes of Places365-Challenge. The selection of such scenes followed conversations with Brazilian Federal Police agents experts in CSAM investigation and labeling, which constantly helped with the research. We emphasize, however, that some classes mentioned by the practitioners were not found within Places365-Challenge, such as photographic studio. Thus, we selected the related category of a television studio, for it depicts photo and video cameras and people posing.

Places365-Challenge already provides training and validation splits mapped accordingly. The test split was then generated from a stratified 10% split from the training set, given that the remapping and filtering made for a highly imbalanced dataset. A small sample of the training set can be seen in Figure 5.5 while numeric details on the final dataset produced are shown in Table 5.2.



(a) bathroom (b) bedroom

(c) child's room (d) classroom

(e) dressing room (f) living room

(g) studio (h) swimming pool

Figure 5.5: Example of image samples for each class within the custom Places8 dataset.

Table 5.2: Description of the Places8 dataset. The class represents the final label used, while the original categories stand for the original Places365 labels. Places365 already provides training and validation splits mapped accordingly. The test set comes from a stratified 10% split from the training set.

Class	Test	Training	Validation	%	Original Categories
bathroom	5740	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	4650	41,849	300	10.8	child's room, nursery, playroom
classroom	3751	33,763	200	8.7	classroom, kindergarden classroom
dressing room	2432	21,889	200	5.7	closet, dressing room
living room	9940	89,458	500	28.7	home theater, living room, recreation room, television room, waiting room
studio	1404	12,633	100	3.3	television studio
swimming pool	1505	13,547	200	3.5	jacuzzi, swimming pool
Total	40,534	364,806	2300	100	

Indoors.all & Indoors.real

We proposed two pretraining datasets for self-supervision: Indoors.all and Indoors.real. The former comprises all chosen Hypersim [107], OpenRooms [79], InteriorNet [77], and Places365-

Challenge [134] indoor images, while the latter is then simply the Places365-Challenge indoor images. Indoors.all contains 7,778,152 images and Indoors.real 2,227,439 images, both used in contrastive pretext tasks.

For comparison, Indoors.all contains over 6 times as much images as ImageNet-1k[2]. This fact led us to reduce the number of epochs so that the total number of images the model sees in the pretraining part is near equal with the model zoo ImageNet pretrained models, which were mostly pretrained for 100-200 epochs. Therefore, Indoors.all SSL pretraining was executed with 25 epochs, and Indoors.real with 50 epochs.

However, we emphasize Indoors.all contains not only real but also synthetic images and their rendered views (depth, segmentation, etc.). These views are used as pseudolabels in the SSL pretext task, which we believe is more informative than random augmentations and could potentially produce better models.

5.1.3 Litmus Test Dataset

Beyond simply checking balanced accuracy, responsible machine learning must also understand the limitations and flaws of produced models [43, 44]. In particular, this work aims at helping people in general, considering the various possible demographics and social backgrounds from which data can come. This is not a simple task in most cases, and it is virtually impossible to find all deficiencies, but we can focus on a few specific, and well-defined features [32].

The machine learning model planned for this Master's thesis will be used for CSAM investigation, but one should notice that it can serve other purposes. We assess its level of generalization with a small custom dataset that considers some underrepresented features from the Places8 dataset as means of testing, auditing, and even understanding the models produced. This step does not require large amounts of data, but different enough imagery is arranged to break the classifier and expose its most serious flaws.

Thus, we produce a small "litmus dataset" from online images to check if the model performance holds outside of the controlled nature of Places8. The dataset comprises 10 images per class, with the 8 original Places8 classes: bathroom, bedroom, child's room, classroom, dressing room, living room, studio, and swimming pool.

The Litmus Test dataset is a sample of images taken from Google images, Bing images, and the Dollar Street dataset [108]. All images have a free license to share, modify and use, including Dollar Street, licensed under CC-BY 4.0 Commercial.

Dollar Street is an annotated image dataset of 289 everyday household items photographed from 404 homes in 63 countries worldwide. It contains 38,479 pictures, split among abstractions (image answers for abstract questions), objects, and places within a home. This dataset explicitly depicts underrepresented populations and is grouped by country and income. Not all countries are present, but there is a balanced amount of pictures per region, and most images come from families who live with less than USD $1000 per month [108].

The dataset (Figure 5.6) is split into folders with the label name, but an accompanying metadata YAML [3] file identifies each image with an id, file path, source, URL, and label. In that

[2]ImageNet-1k contains 1,281,167 training images, 50,000 validation images, and 100,000 test images

[3]YAML was selected for its better readability and is user-friendliness in comparison to other used metadata schemas, such as JSON

regard, we selected all images using the label keyword in the Google/Bing search engine or Dollar Street folder (when possible, given that Dollar Street does not contain all Places8 classes). However, they were chosen, so there was a mix of simple and challenging classifications.

(a) bathroom (b) bedroom

(c) child's room (d) classroom

(e) dressing room (f) living room

(g) studio (h) swimming pool

Figure 5.6: Example of image samples for each class within the custom Litmus Test dataset.

We emphasize that this is a dataset aimed at inference with a simple check, and no training is performed. A complete datasheet of the Litmus Test dataset's information can be found in Appendix A.

5.2 Experimental Design

Following the pipelines presented in Chapter 4, we conducted a sequence of pretraining and finetuning experiments to find the best model for the indoor scene classification task (our target task).

We explored four SSL techniques for comparison against a full supervised baseline transferred from ImageNet. We selected ResNet-50 ($1\times$) as the main backbone for all experiments, given the wide adoption of such neural network architecture on both supervised and self-supervised learning and the fact most released pretrained models are available with this architecture.

We carried out the experiments on the Places8 dataset as our target task. Therefore, the main results presented and the models built are centered and hyperoptimized around this particular task.

In this context, all experiments share a few constant parameters used throughout the pretraining and finetuning steps (Table 5.3). These values mostly come from settings and recommendations from other machine learning and self-supervision community researchers.

Table 5.3: List of most essential constants used for all experiments targeting the Places8 dataset.

Parameter	Values
batch size per GPU	1024
image size	224×224 pixels
network backbone	ResNet-50
AMP enabled	true
AMP settings	PyTorch mixed precision default
activation checkpointing enabled	true
activation checkpointing settings	2 splits
optimizer	LARS
optimizer clip gradient enabled	true
optimizer LARS epsilon	1e-08
optimizer LARS trust coefficient	1e-03
optimizer weight decay	1e-06
optimizer momentum	0.9
optimizer Nesterov enabled	true
optimizer regularize batch normalization	false
optimizer regularize bias	true
learning rate auto-scaling enabled	true
learning rate base batch size	1024
number of epochs	100
early stopping warmup range	5
early stopping min delta	0.1
early stopping patience	5
number of cross-validation folds	5

All experiments were set to run for 100 epochs. The validation loss with early stopping is used to determine when the training has reached its optimal point. Our early stop procedure started after the first 5 epochs and checked at every epoch if the validation loss had raised beyond 0.1 from the minimum seen value or patience of 5 epochs. Early stopping was fundamental to reach the best possible models with minimal wasted training time.

We applied a 5-fold cross-validation classification protocol. It gives reliable results and enough points for comparison while keeping the number of experiments in a feasible state.

5.2.1 Pretrained Models

Several experiments had the model weight initialization from other pretrained neural networks, which other researchers released. Starting from pretrained weights generally allows for better final representations while exempting us from having to train any ImageNet models.

In this regard, all pretrained models gathered were trained on the ImageNet-1k with a ResNet-50 architecture and using a temperature equal to 0.1, shown in Table 5.4. For the sake of convenience, we fetch the model with the highest ImageNet top-1 score for each technique, but it is essential to notice that pretrained models starting from 100 epochs and various batch sizes are released, all with quite close accuracy metrics. However, some selected models have accuracies different than those published (Table 2.1), as these models were not always made publicly available or fully compatible with our deep learning tools.

In light of that and given the time constraints, all SSL pretraining made by ourselves occurs with no more than the equivalent image amount as 100 ImageNet epochs[4].

Table 5.4: Pretrained models used for weight initialization on some of the SSL pretraining and model finetuning experiments. Accuracy stands for ImageNet-1k top-1 accuracy.

Method	Pretrain Epochs	Pretrain Batch Size	Accuracy (%)	Source
Supervised	–	–	75.45	VISSL Model Zoo
SimCLR	800	4096	69.68	VISSL Model Zoo
SwAV	800	4096	74.92	VISSL Model Zoo
BarlowTwins	1000	2048	71.80	VISSL Model Zoo
SupCon	1000	1024	79.10	SupContrast

5.3 Results

5.3.1 Supervised Baseline

Initially, we carried out a sequence of finetuning experiments from an ImageNet supervised backbone to find the optimal supervised baseline for our main target task. The goal is to make an extensive comparison when looking for the best possible model, so this baseline is, later on, going to be compared with the SSL models. We optimized this supervised pipeline more thoroughly to make it more challenging than the SSL counterparts.

We applied a 5-fold cross-validation end-to-end finetuning from a ResNet-50 backbone pretrained in ImageNet with a supervised pipeline. This set of experiments explored different learning rate scenarios to find the best model for the Places8 dataset.

Some critical constants and parameters used in these experiments are shown in Table 5.5. The hyperparameterized column in this table defines which parameters were tried in the grid search procedure and all values checked are separated by commas. Beyond that, each experiment was repeated four times to diminish the possible effects of shuffling during the training and get a more stable result.

The best-supervised model gets $84.9\% \pm 2.8\%$ average accuracy on Places8 and uses a base learning rate of 0.001 with the cosine half wave $3\times$ restarts scheduler and batch size per GPU of 1024. There is a significant dependency of the performance with the learning rate value, as shown in Figure 5.7(a), where the impact of the lower learning rate is visible.

Figure 5.7(a) is a violin plot of all experiments made with the supervised learning model grouped by the initial learning rate value. The violin plot is a distribution plot, which shows if

[4] https://github.com/facebookresearch/vissl/blob/main/MODEL_ZOO.md

Table 5.5: List of constants and hyperparameters used in the supervised baseline experiments in the Places8 dataset.

Parameter	Values	Hyperparameterized
augmentations	horizontal flip	
network head	MLP 2048×8	
learning rate base value	0.1, 0.01, 0.001	yes
learning rate scheduler	linear warmup + cosine half wave 3x restarts, linear warmup + cosine, linear warmup + constant with 3 decays	yes
optimizer	SGD, Adam, LARS	yes
batch size per GPU	64, 256, 1024	yes

the data points are equally distributed around the mean or if there are possibly multiple clusters. When performing multidimensional hyperparametrization and k-fold cross-validation, the violin plot is a helpful tool to find possible imbalances in the data while also allowing for a quick comparison analysis. Particularly in Figure 5.7(a), each violin is composed of 20 data points because each of the 5 folds is repeated four times.

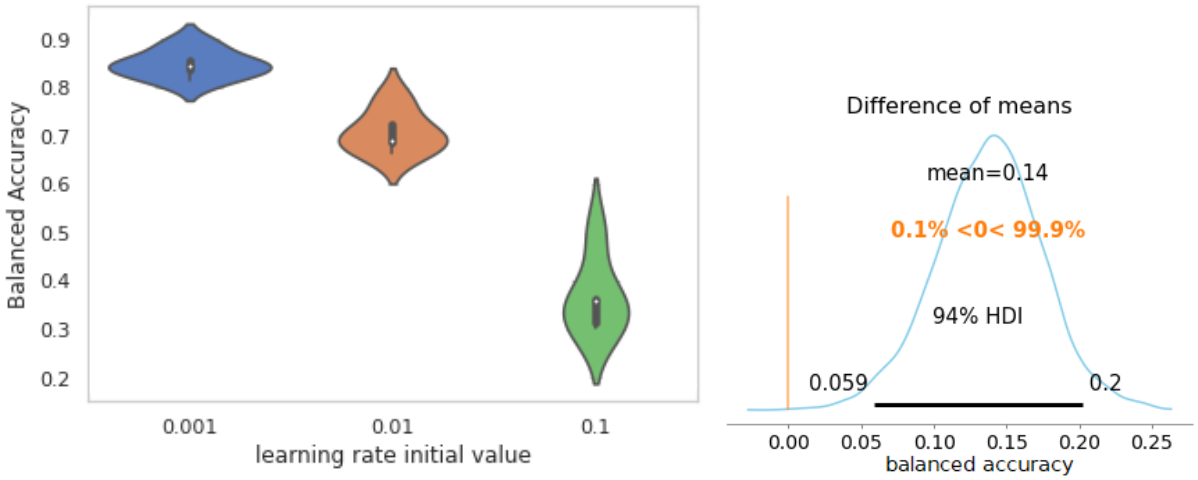
Table 5.6: Hyperparameters for the best-supervised model.

Hyperparameter	Value
learning rate base value	0.001
learning rate scheduler	linear warmup + cosine half wave 3x restarts
optimizer	LARS
batch size per GPU	1024

The plot in Figure 5.7(b) is the output of the BEST analysis explained in Section 2.4. It shows the fitted gaussian distribution for the difference of means among the balanced accuracy of two different models. The orange vertical line is a reference for the value of zero, and the orange headers signal how much of the plotted area is before and after this reference. The fact that over 99.9% of the distribution lies after zero implies a 99.9% confidence rate that 0.001 is superior to 0.01 as a learning rate. Beyond that, the value at "mean" represents the average difference of balanced accuracies, so the learning rate of 0.001 increases the balanced accuracy of the model by 14 percentage points on average when compared to the learning rate of 0.01. Finally, the High-Density Interval (HDI) indicates the interval of values where 94% of the distribution mass lie, similarly to a confidence interval [71].

In particular, the results were statistically equivalent for the other hyperparameters tested, so the LARS optimizer and batch size per GPU values were selected to minimize the training time while maintaining good model performance. Beyond that, considering the large sizes of the datasets used, this batch size is still far from the noise scale and, together with the usage of the LARS optimizer, represents minimal chances for a generalization gap [55, 87]. The batch size of 1024 per GPU, LARS optimizer, and learning rate scheduler of cosine half wave 3× restarts were kept the same for all remaining experiments.

Beyond that, we are also interested in understanding how ambiguous the classes are among themselves for the trained model, and the normalized confusion matrix is an excellent tool to



(a) (b)

Figure 5.7: Results for the supervised learning model finetuned on Places8. (a) Violin plots of balanced accuracy versus the different initial learning rates. (b) BEST difference of means with learning rate base values of 0.001 and 0.01.

capture such kind of trouble. Figure 5.8 shows the most significant confusion between the bedroom and living room (11.8%), child's room and classroom (10.2%), dressing room and bathroom (8.2%), and even child's room and bedroom (6.9%). On the other hand, the bathroom, swimming pool, and studio seem more easily distinguishable for this model overall.

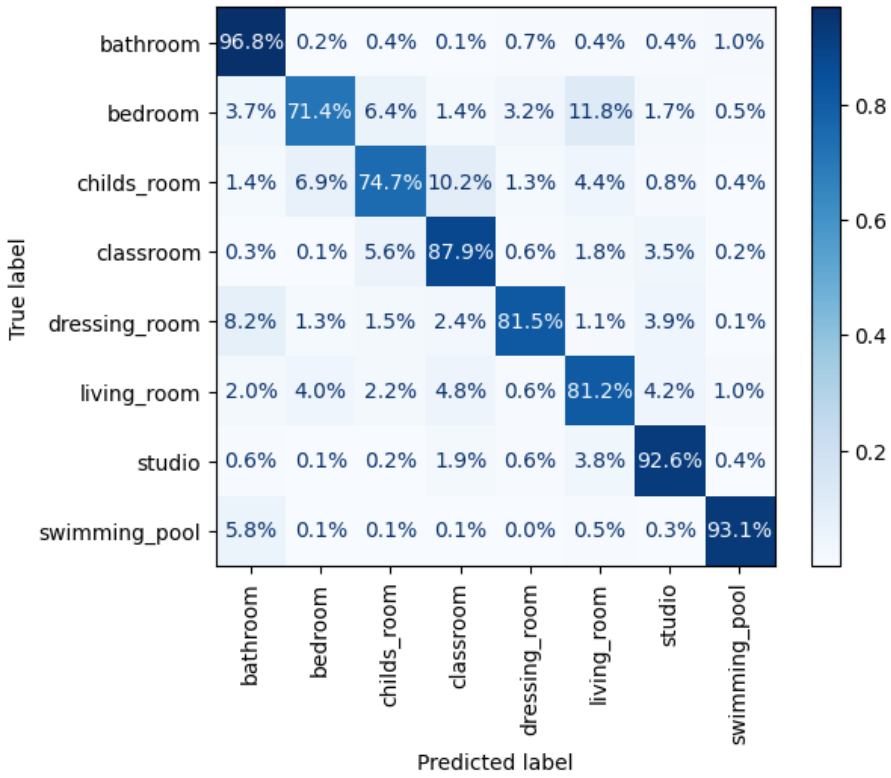


Figure 5.8: Confusion matrix for the best-supervised model finetuned on Places8 with values normalized by a number of elements in each class.

5.3.2 SSL Hyperparametrization

These initial SSL experiments intend to understand the best settings for self-supervision. The goal is to find the best optimizer configurations, such as pretraining and finetuning learning rates, that will be repeated for the other SSL techniques evaluated.

These SSL experiments attempt to find the best-pretrained model by starting from the same settings used in the literature and varying some hyperparameters to find the best for our target task. However, it is essential to notice that there is no way to evaluate an SSL pretrained model in isolation, only by finetuning on a downstream task. Therefore, each one of the models produced on this step was further hyperoptimized at cross-validation model finetuning.

Along this line, the finetuning model experiments also started from settings similar to those found in the literature, with minor variations.

To such end, the SwAV technique was chosen for this particular set of experiments given it has been verified as the most reliable on ImageNet similar data [36] and, as mentioned on Chapter 2, it is more efficient in a distributed training environment and gives better results even with small batch sizes.

Moreover, we used an ImageNet pretrained SwAV model released by Meta AI as one of the options for weight initialization on both SSL pretraining and finetuning. This means that some models were pretrained from scratch on scene-centric datasets while others were pretrained from object-centric models, and also that some models were finetuned without any scene-centric pretraining, i.e., were finetuned direct from the ImageNet SwAV model.

Consequently, this step helps us to understand the best average conditions for self-supervised learning and reduces the total number of experiments.

In short, this set of experiments relied not only on the SwAV technique but also involved pretraining with two different datasets, Indoors.all and Indoors.real, and using two different weight initialization techniques. Among other hyperparameters tried, these let us understand how SSL algorithms best specialize in the scene recognition domain. Other training settings were kept equal to the supervised baseline.

In that sense, Tables 5.7 and 5.8 show all constants and parameters hyperoptimized for SSL pretraining and 5-fold cross-validation end-to-end finetuning, respectively. Similar to the supervised baseline, grid search was used, and each experiment was repeated four times (two times for each SSL pretraining and two times for each model finetuning).

Moreover, the two chosen pretraining datasets had far different sizes, so the total number of epochs was balanced off a 100 epochs pretraining on ImageNet following the Equation 5.1. In short, the number of epochs for Indoors.all and Indoors.real were rounded to 25 and 50, respectively. This allows the models to be more fairly compared, as they would be exposed to approximately the same number of images. On the other hand, the limit of 100 ImageNet epochs was chosen exclusively out of time constraints, for each SSL pretraining took around 30 hours.

$$\text{dataset epochs} = \frac{\text{ImageNet size}}{\text{dataset size}} * 100 \text{ epochs} \tag{5.1}$$

The best SwAV pretrained model reached 80.0% $\pm$ 1.6% average accuracy on Places8 and used a learning rate pretraining and finetuning base value of 0.01, with the Indoors.real dataset and was initialized from the SwAV ImageNet pretrained model, as shown in Table 5.9.

Table 5.7: List of constants and hyperparameters used in the SwAV hyperparametrization experiment during self-supervised pretraining step. Some of these parameters are specific to the SwAV technique.

Parameter	Values	Hyperparameterized
multicrop	2×160 pixels + 4×96 pixels	
augmentations	horizontal flip, color distortion, gaussian blur	
network head	MLP 2048×2048×128	
temperature	0.1	
number of clusters (SwAV specific)	3000	
learning rate pretraining base value	0.01, 0.001	yes
learning rate scheduler	linear warmup + cosine half wave 3× restarts	
pretrain dataset	Indoors.real, Indoors.all	yes
weight initialization	glorot, ImageNet pretrained model	yes

Table 5.8: List of constants and hyperparameters used in the SwAV hyperparametrization experiment during the cross-validation model finetuning step.

Parameter	Values	Hyperparameterized
augmentations	horizontal flip	
network head	MLP 2048×8	
learning rate finetuning base value	0.01, 0.001	yes

Table 5.9: Hyperparameters for the best SwAV pretrained model.

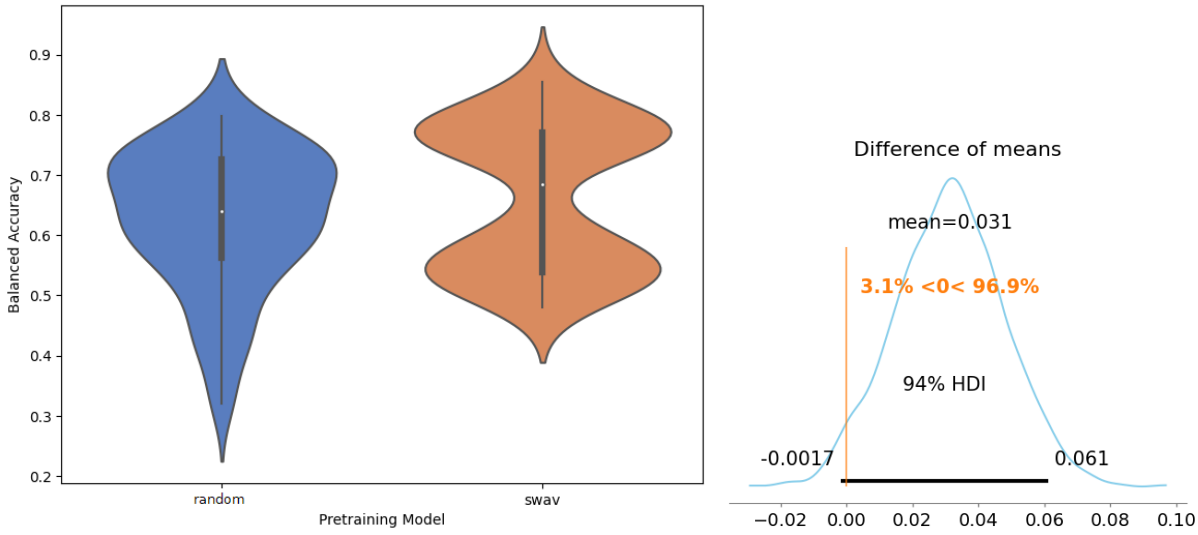
Hyperparameter	Value
learning rate pretraining base value	0.01
learning rate finetuning base value	0.01
pretrained dataset	Indoors.real
weight initialization	ImageNet pretrained model

In this regard, we aim to thoroughly understand the impact of the hyperparameters in this set of experiments.

Weight Initialization

Weight initialization stands for the option of using another model pretrained weights as a starting point. Specifically, we compared the glorot initialization with an ImageNet pretrained SwAV version.

The violin plots in Figure 5.9(a) show SwAV to be a slightly better option in terms of the mean balanced accuracy, but there is clearly a bimodal distribution. Based on the difference of means from the BEST methodology in Figure 5.9(b), we found SwAV initialization positively impacted the balanced accuracy with 96.9% confidence and an average increase of 3.1% compared to the random initialized models.



(a) (b)

Figure 5.9: Results for the SwAV pretrained models finetuned on Places8. (a) Violin plots of balanced accuracy versus the different weight initializations. (b) BEST difference of means between the SwAV pretrained models with random and ImageNet pretrained initializations.

Pretraining Dataset

We analyzed each pretraining dataset's impact to disambiguate the pretraining scenarios more meticulously. For that matter, we analyzed the experiments as having three possible pretraining datasets: Indoors.all, Indoors.real and ImageNet, where the ImageNet pretraining comes from the SwAV ImageNet pretrained model. Therefore, we have five possible configurations:

1. Indoors.all: randomly initialized model pretrained on Indoors.all for 25 epochs and finetuned on Places8;
2. Indoors.real: randomly initialized model pretrained on Indoors.real for 50 epochs and finetuned on Places8;
3. ImageNet: SwAV ImageNet initialized model directly finetuned on Places8 (no second pretraining);
4. ImageNet + Indoors.all: SwAV ImageNet initialized model with a second pretraining step on Indoors.all and then finetuned on Places8;
5. ImageNet + Indoors.real: SwAV ImageNet initialized model with a second pretraining step on Indoors.real and then finetuned on Places8.

The violin plots in Figure 5.10 show all random initialized models had some collapsed models (balanced accuracy < 0.4), whereas the models with ImageNet initialization did not. Despite that, it is not possible to determine the best pretraining dataset configuration.

Based on the difference of means from the BEST methodology in Figure 5.11(a), we verify that the Indoors.real second step versus the Indoors.all second step positively impacted the balanced accuracy with 62.3% confidence and an average increase of 1%.

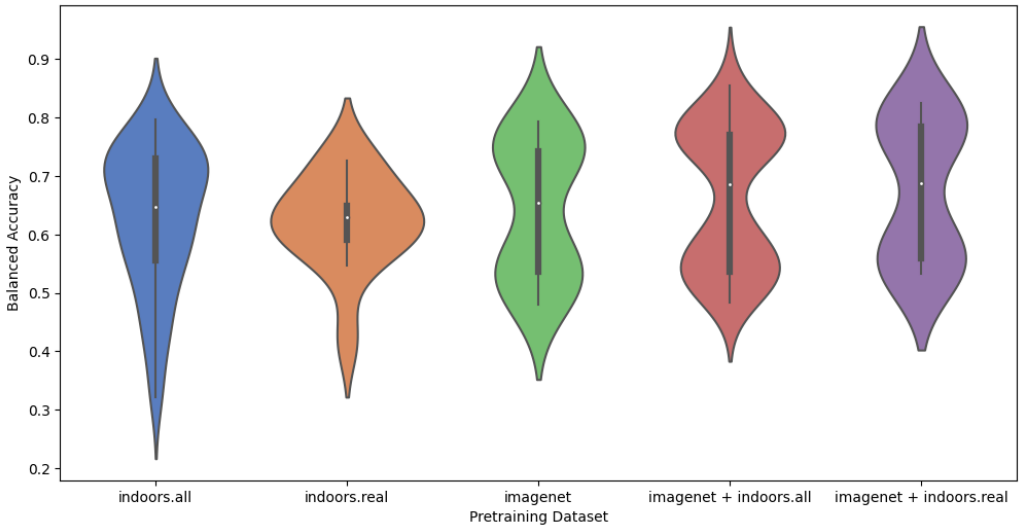


Figure 5.10: Violin plots of balanced accuracy versus the different pretrain datasets configurations tried for the SwAV pretrained models finetuned on Places8. The presence of the ImageNet dataset means the weights were initialized from a SwAV model pretrained on ImageNet.

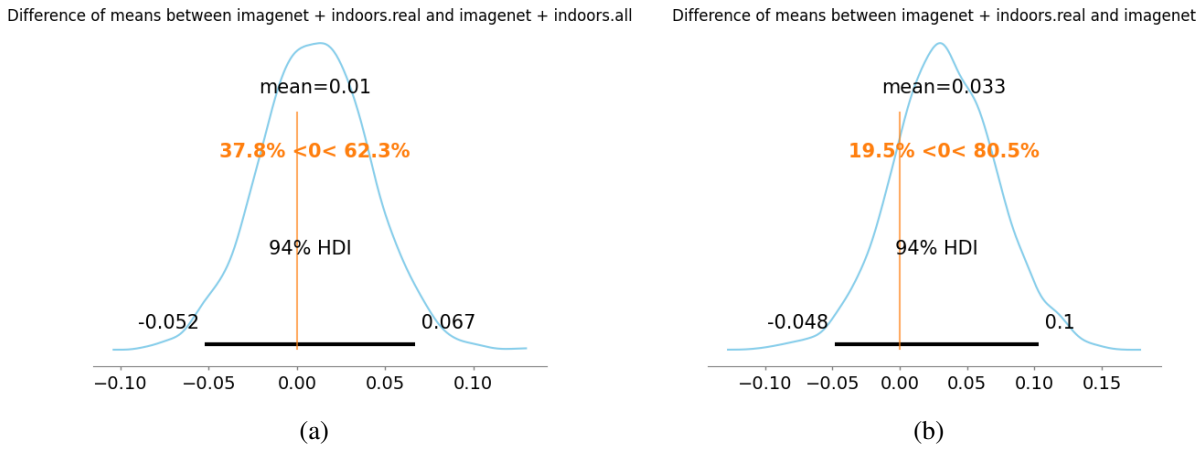


Figure 5.11: BEST difference of means between the SwAV pretrained models with a second step on (a) Indoors.real versus Indoors.all, and (b) Indoors.real and without it.

This result alone shows it is pretty hard to say Indoors.real is better than Indoors.all as a better second pretraining dataset. All in all, they could most likely be equivalent.

With that in mind, we question if there is any impact on adding this costly second pretraining step. We observe then in Figure 5.11(b) that an Indoors.real addition positively impacted over ImageNet alone with 80.5% confidence and an average increase of 3.3%, which is definitely more significant.

Learning Rates

We analyzed the impact of different pretraining and finetuning learning rates as a final comparison. The violin plots in Figure 5.12 show the higher learning rates to be a better option in terms of the mean balanced accuracy on both pretraining and finetuning.

Finally, we question which learning rate had the higher impact on the final model perfor-

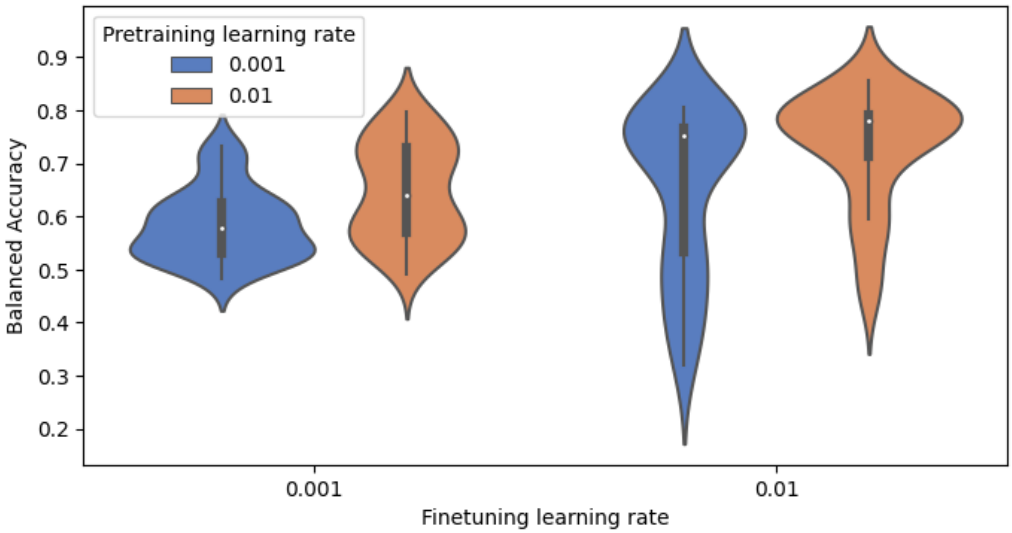


Figure 5.12: Violin plots of balanced accuracy versus the different pretraining and finetuning learning rates configurations tried for the SwAV pretrained models finetuned on Places8.

mance. For that matter, once more, we employed the BEST analysis, showing that, on average, the pretraining learning rate increase led to a 10 percentage points improvement in balanced accuracy while the same finetuning learning rate increase led to a 7.2 percentage points improvement (Figure 5.13).

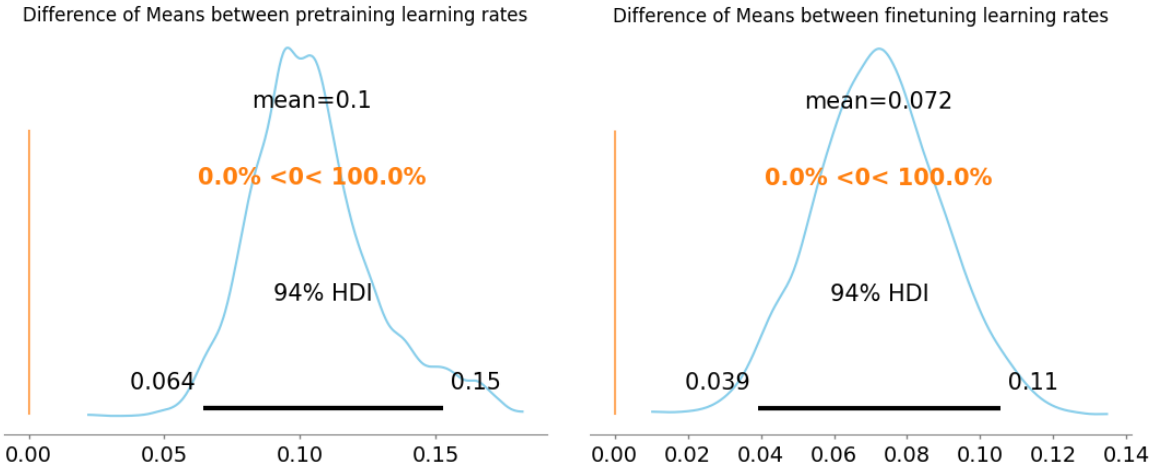


Figure 5.13: BEST difference of means between (a) pretraining learning rates and (b) finetuning learning rates for the SwAV pretrained models finetuned on Places8.

Confusion Matrix

In summary, we have the higher learning rates as a confident better choice for SwAV on Places8, which is in opposition to the results from the supervised learning. Plus, employing an ImageNet pretrained model and a second pretraining step with a scene-centric dataset was beneficial, even though the best scene-centric dataset could not be determined with high confidence.

Beyond that, we are also interested in understanding how ambiguous the classes are among themselves for this SSL model. The normalized confusion matrix in Figure 5.14 shows overall higher error rates values for the same pair of classes we had in the supervised experiment: bedroom and living room (19.1%), dressing room and bathroom (18.0%), bedroom and child's room (14.4%), and even classroom and child's room (14.1%). In particular, several classes seem to be confused with living room for this model, which is likely an effect of the high diversity in this class, given it is a union of multiple Places365 classes. On the other hand, swimming pool still seems to be more easily distinguishable for this model.

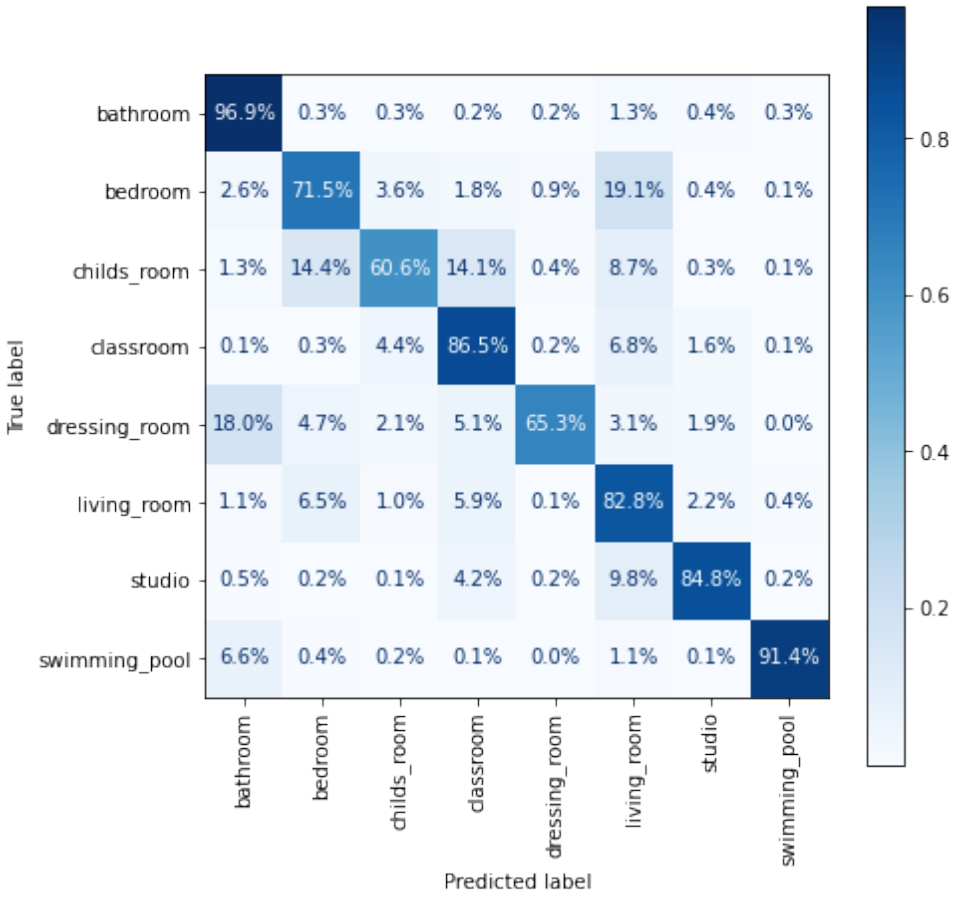


Figure 5.14: Confusion matrix for the best SwAV pretrained model finetuned on Places8 with values normalized by a number of elements in each class.

5.3.3 Self Supervised Technique Comparison

Beyond SwAV, other SSL techniques can be tried out on the target task. We selected one technique to find the best hyperparameters and reused them on others primarily due to time constraints: grid-search hyperparametrization and SSL pretraining are expensive enough that we cannot repeat said experiments for all other SSL techniques.

With that in mind, we chose SimCLR [24], BarlowTwins [128] and Supervised Contrastive (SupCon) [65] as extra SSL methods to be tried out. The option for these specific techniques was due to the availability of public ResNet-50 models and the level of performance on ImageNet and ImageNet similar datasets [36].

In particular, SupCon [65] can leverage classes to enhance the SSL pipeline, and it was selected for this Master's thesis as we hypothesize that the usage of the datasets categories could improve the quality of the final representations. With this set of experiments, we intend to find the best SSL model for our target indoor recognition task.

Thus, given that a set of hyperparameters was obtained for the SwAV technique, we pretrained and finetuned the other self-supervised models to check if a different SSL technique could provide better results on Places8. This set of experiments follows the same steps described in the previous sections, and Table 5.10 shows all constants and parameters specific to each SSL technique.

Table 5.10: List of constants and hyperparameters used in the self-supervised pretraining step. Some of these parameters are technique specific.

Parameter	Values
temperature (all SSL)	0.1
embedding dimension (all SSL)	128
number of clusters (SwAV)	3000
lambda (BarlowTwins)	0.0051
scale loss (BarlowTwins)	0.024

Moreover, we still pretrained on our two datasets, Indoors.all and Indoors.real, to improve the confidence level to assess better the dependency of different pretraining options on the downstream task.

In that sense, we initialized all models with the ImageNet pretrained versions (there was no cross pretraining, i.e., no BarlowTwins pretraining happened on, for example, the SimCLR ImageNet model or vice-versa). The training from scratch was not tried in this batch of experiments because we observed in the last section that there is a clear and definite advantage to starting from ImageNet.

Even including the SwAV benchmarks, the best SSL-based model found gets 86.3% $\pm$ 2.4% average balanced accuracy for Places8 with the BarlowTwins technique pretrained on the Indoors.real dataset and initialized from the BarlowTwins ImageNet pretrained model, as shown in Table 5.11.

Table 5.11: Hyperparameters for the best SSL pretrained model.

Hyperparameter	Value
SSL technique	BarlowTwins
pretrain dataset	Indoors.real
weight initialization	ImageNet pretrained model

Similar to the last section, we aim to thoroughly understand the importance of these parameters on the Places8 classification task.

SSL Technique

Initially, we compare the differences between the different self-supervision techniques. The violin plots in Figure 5.15 show BarlowTwins to clearly be a better option in terms of the mean

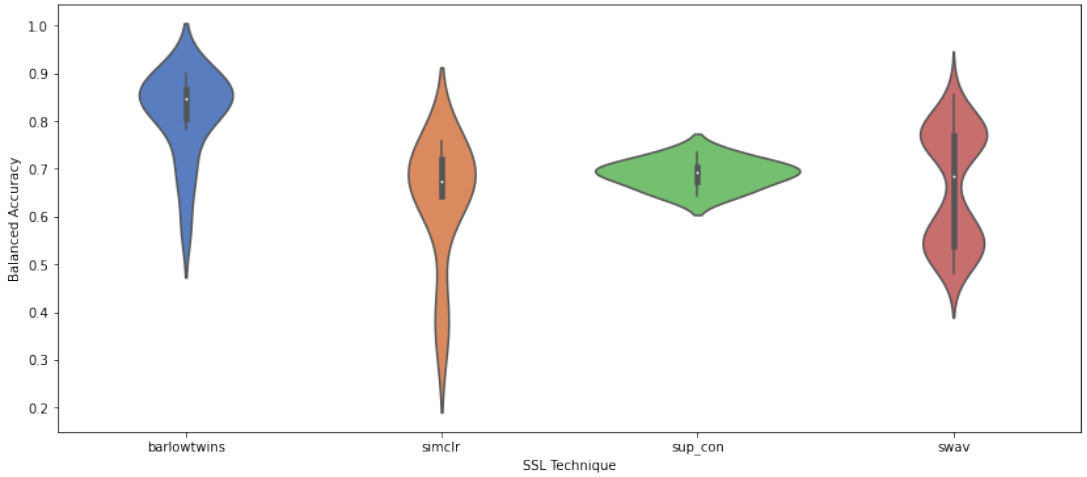


Figure 5.15: Violin plots of balanced accuracy versus the different SSL technique models finetuned on Places8.

balanced accuracy.

Pretraining Dataset

To confidently decide which pretraining case is better, we again compare the influence of the pretraining dataset on the final downstream performance. Since only ImageNet pretrained models are used on the weight initialization, there are now three possible dataset configurations:

1. ImageNet: ImageNet initialized model directly finetuned on Places8 (no second pretraining);

2. ImageNet + Indoors.all: ImageNet initialized model with a second pretraining step on Indoors.all and then finetuned on Places8;

3. ImageNet + Indoors.real: ImageNet initialized model with a second pretraining step on Indoors.real and then finetuned on Places8.

The violin plots in Figure 5.16 show there were some collapsed models (balanced accuracy < 0.4) among the ImageNet only set, whereas the models with a second pretraining step on an indoor dataset never displayed such behavior.

Based on the difference of means from the BEST methodology in Figure 5.17(a), we verify the Indoors.real second step had a more positive impact on the balanced accuracy than Indoors.all with 88.7% confidence and an average increase of 3%.

Compared with the results from the last section, it is now possible to say with actual confidence Indoors.real is a better second pretraining step than Indoors.all. Thus, using synthetic images along with real ones was not beneficial to the final performance on Places8, even though the total number of images was much higher. We hypothesize this problem could be due to the difference in epochs between Indoors.real and Indoors.all (which is more thoroughly analyzed in the ablation study in Section 5.3.4). However, the presence of synthetic views (depth, segmentation map) in the dataset most likely disturbed the self-supervised learning process, given

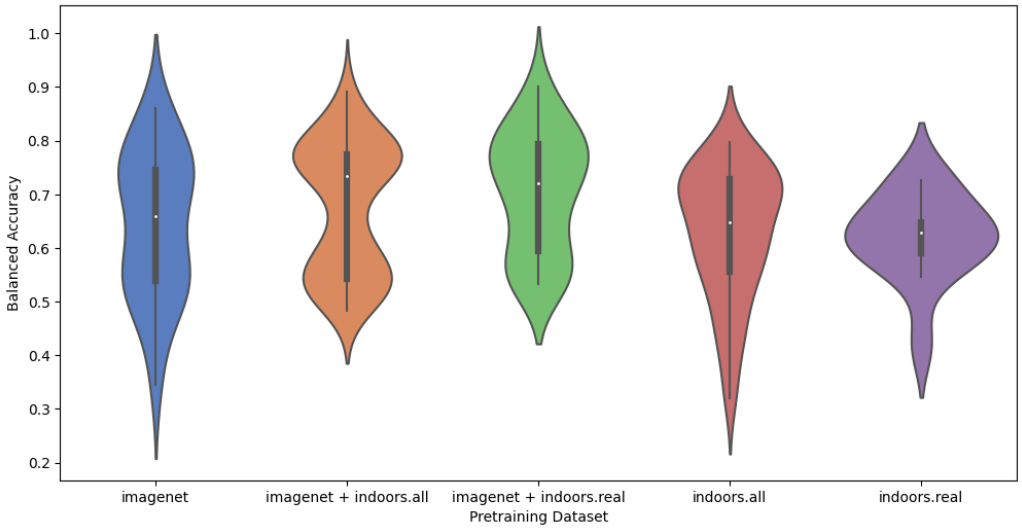


Figure 5.16: Violin plots of balanced accuracy versus the different pretrain datasets configurations tried for the SSL pretrained models finetuned on Places8. The presence of the ImageNet dataset means the weights were initialized from a model pretrained on ImageNet.

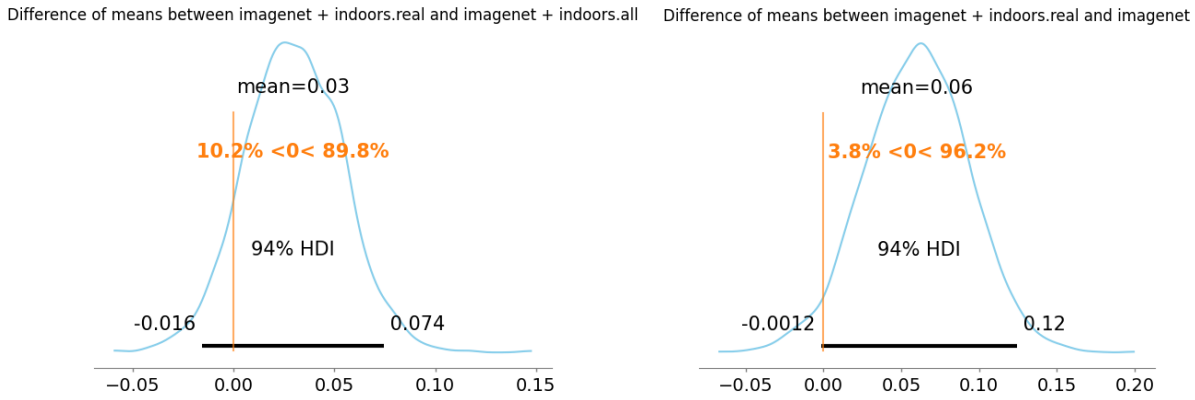


Figure 5.17: BEST difference of means between the SSL models with a second step on (a) Indoors.real versus Indoors.all, and (b) Indoors.real and without it.

there were no similar images in the downstream task (Places8 has no images like those). Then, it might be useful to attempt a new indoor dataset containing only photorealistic figures.

Again, we question if there is any impact at all on adding this costly second pretraining step. We notice then in Figure 5.17(b) the experiments made show an Indoors.real addition had a positive impact over ImageNet alone with 96.2% confidence and an average increase of 6%, which is definitely significative for machine learning, especially on a hundred thousand image dataset like Places8.

5.3.4 Ablation Study

Any statistical study is filled with multiple coupled components and hyperparameters influencing the model performance in complex ways. It is thus essential to investigate the performance

of an AI system by removing or altering specific parts in isolation to understand its contribution to the whole system.

Toward the best overall model, we designed some particular experiments to understand better the behavior of the created models under different training conditions.

In this section, we aim to discover how SSL pretraining affects the final model performance. We verify how different pretraining datasets and the number of pretraining epochs influence the final model performance. Considering the SSL pretraining step is the most expensive part of the pipeline, it is helpful to find the best conditions and limitations of this step, so researchers can maximize the performance while minimizing the costs.

Pretraining Epochs

Given the expensive nature of SSL pretraining, we finetune separate models that have been pretrained with BarlowTwins on our pretraining datasets for 1, 2, 3, 5, 10, 20, 30, 40, and 50 epochs and report the top-1 accuracy on Places8, as shown in Figure 5.18.

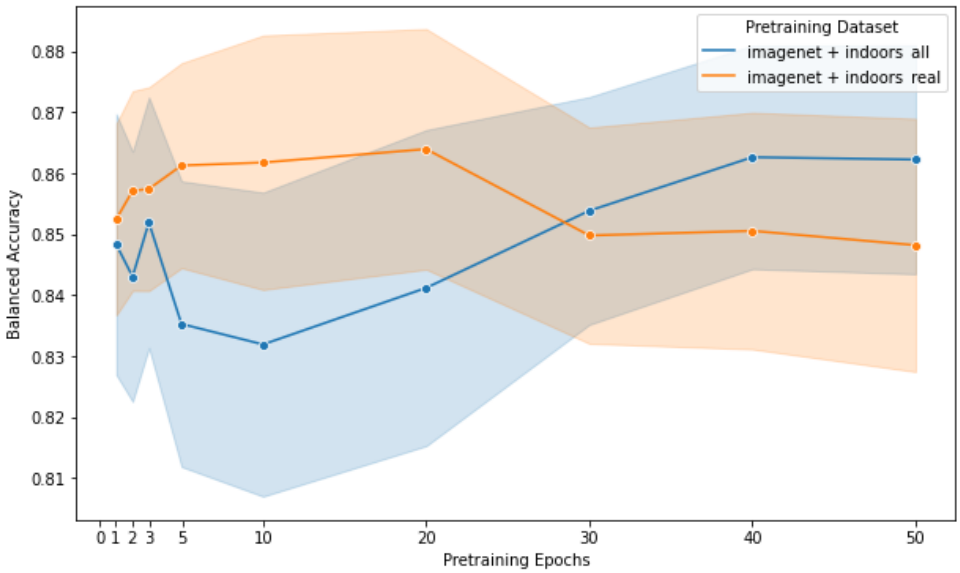


Figure 5.18: Lineplot of top-1 balanced accuracy on Places8 versus the number of pretraining epochs on each pretraining dataset for BarlowTwins initialized from an ImageNet pretrained model. The bands around the line represent the standard deviation on all experiments done for one particular epoch.

The results show overall Indoors.real converges quicker than Indoors.all, which takes 50 epochs to reach performance that the first reached on 20 epochs. On the other hand, we notice Indoors.real performance decreases in later epochs. This experiment starts from an ImageNet pretrained model, so the conclusions will likely differ if starting from scratch.

5.3.5 Final Results on Places8

A summary of all experiments conducted in these previous step is found in Table 5.12 and, together with Figure 5.19, we see the best SSL model (Table 5.11) performs better on Places8

than the best-supervised counterpart (Section 5.3.1) with 98.7% confidence and, on average, shows a 2.2% increase in the balanced accuracy.

Therefore, we can confidently affirm that an SSL model should be used for this task and proceed with training/test this model, specified in Table 5.11, on the Places8 dataset, without cross-validation.

Table 5.12: Summary of results for each attempted technique

Technique	Hyperparameters	Experiments	Best Model Average Balanced Accuracy
Supervised	4	540	84.9% ± 2.8%
SwAV	4	320	80.0% ± 1.6%
SimCLR	2	160	73.9% ± 4.4%
SupCon	2	160	77.3% ± 0.7%
BarlowTwins	2	160	86.3% ± 2.4%

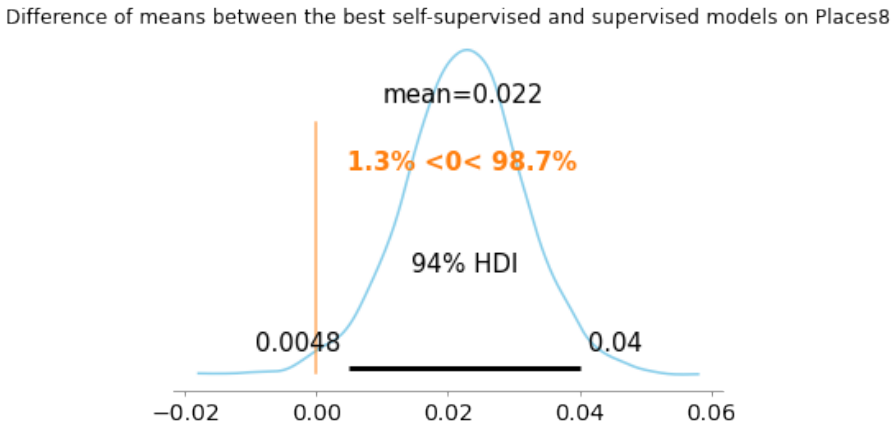


Figure 5.19: BEST difference of means between the best SSL and supervised models. The BEST model is input with the balanced accuracy of each cross-validation fold test.

For the final evaluation, the model was finetuned on all Places8 training sets and evaluated on the test set, which was held out up to this point, following the pipeline from Section 4.2. The results showed 71.6% balanced accuracy on this task, which is expectedly lower than the average balanced accuracy on cross-validation, but it is still an acceptable result considering scene classification often demonstrates low accuracies on the Places dataset [129].

In spite of that, when we look at the confusion matrix in Figure 5.20, the differences among the classes show how the test set can be much more challenging than cross-validation. Here “dressing room” is the best-performed class, with “bathroom” and “living room” following in terms of true positives. However, this model clearly avoids classifying images as “bedroom” or “classroom”, in opposition to what has been seen in the cross-validation phase, as in Figure 5.14 for example. Overall, the model has unbalanced performance yet reaches satisfactory results in most classes.

5.3.6 Litmus Test

The model inferred on 80 images, with 62 right and 18 wrong classifications, which represents 77.5% accuracy, higher than our final reported test accuracy on Places8, which is an acceptable

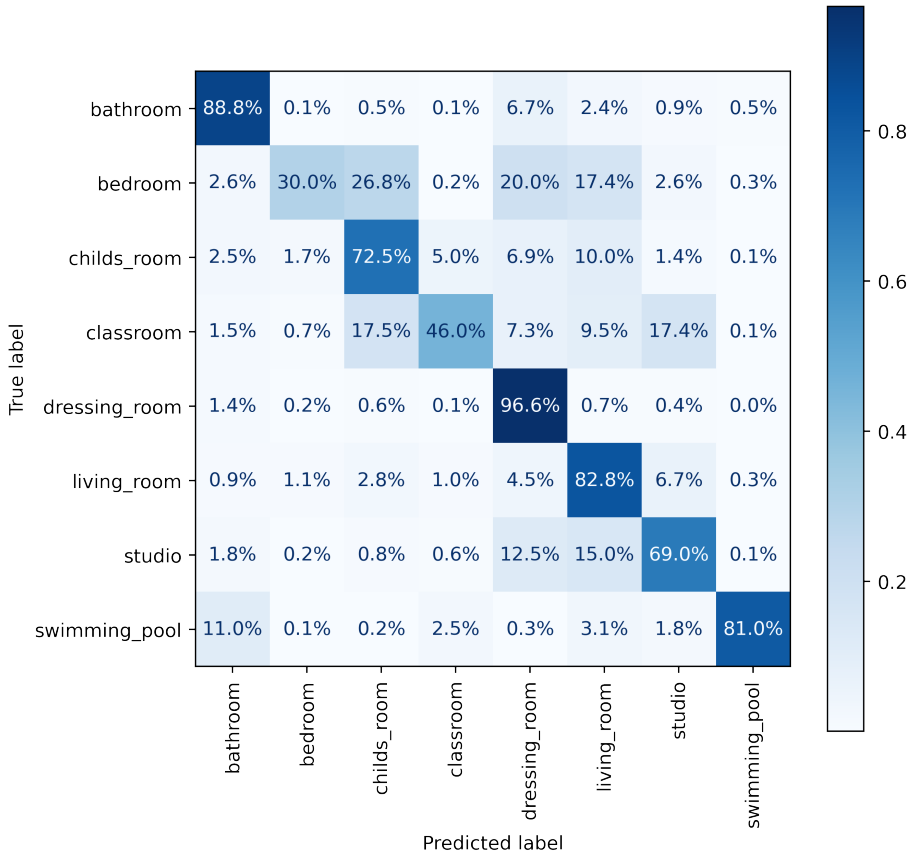


Figure 5.20: Confusion matrix for the best model on Places8 test set with values normalized by a number of elements in each class.

result when we look at the images within the litmus dataset.

First, "classroom" is the class with the best overall performance, with all instances correctly classified and no false positives, i.e., no other image was misclassified as classroom (Figure 5.21).

On the other hand, child's room seems to be the most difficult classification, with the model often preferring bedroom instead, even for the "magazine cover" images. However, it seems the model can be fooled to generate false positives when children are present in the image, which should be an expected bias, as shown on the fourth image in the "studio" row in Figure 5.21.

In that sense, even though the Places8 "studio" class was trained from the original "television studio" Places365 category, the model seems to be able to adapt to other kinds of studios, seemingly focusing on the presence of lights or homogeneous backgrounds. However, it cannot correctly identify "studio pictures", a kind of picture commonly present among CSAM.

Beyond that, swimming pool is still correctly classified when depicted outdoors. The water texture may be such a clear sign of the pool that even with unseen traits, such as the sky, it can still classify it correctly. The only exception is when the picture is taken from inside the pool, as shown on the second image in the "swimming pool" row in Figure 5.21, which was selected to trick the classifier.

Finally, there seems to be a high level of confusion between bedroom, living room, and child's room, especially when looking at low light and non-standard images (Dollar Street), which we believe is capable of improvement.

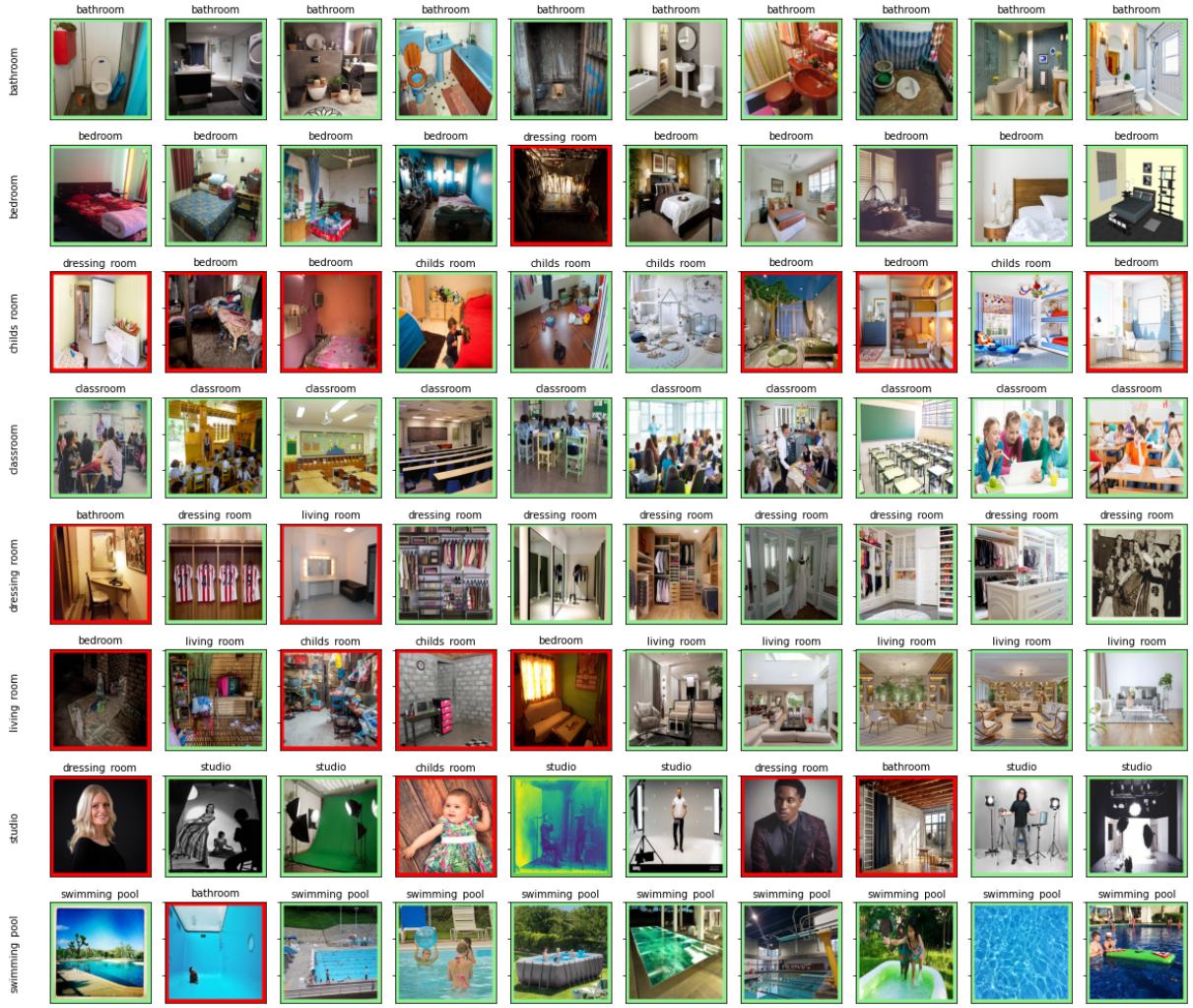


Figure 5.21: Image grid of inference results on the Litmus Test dataset. Each row represents one label, and the name on the left is the true label for each image, while the name above it is the predicted label. A green frame highlights if the prediction matches the true label; otherwise, a red frame is placed on the image.

5.3.7 CSAM Final Test

A Brazilian Federal Police agent evaluated our best scene classifier on CSAM data. We created a script, which will be available in the final repository without sensitive information, that executes the model through all images within a directory and outputs a dataframe (tabular data) with the predicted class for each image. A subset of the results was manually checked to confirm the model's performance on this data. We emphasize that only inference is made in this step, and this Master's thesis performs no training whatsoever with CSAM.

The script was made to run inside a Docker container, and it was accompanied by a deep learning model as a .torch file. All required files were *zipped* together and sent via email to the agent. We emphasize that no CSAM was sent or shared using email or any communication protocol, only the script and the model trained on the Places8 dataset.

In total, 46,006 images had their scenes classified by the model, and the agent manually labeled 615 of those. This subset is grouped into two categories of child abuse material: CSAM and CSAM Suspect. The CSAM category is constituted of levels A and B from Table 1.1,

which are the most serious and sensitive classes. The CSAM Suspect category is thus level C from Table 1.1 and does not show any nudity yet hints of eroticism. Among these, there are 313 images from the CSAM category and 302 images from the CSAM Suspect category.

The histogram with the labeled scenes is depicted in Figure 5.22, in which there are more classes than Places8 was trained to classify. In particular, many images focused on body parts and were thus labeled "undefined".

On a different note, it is clear that some classes are more common in one category than the other, with "studio" being especially prevalent in CSAM Suspect (the agent affirms there is a correlation between this kind of scene and erotic posing). Moreover, the scenes of "classroom" and "dressing room" are not seen in any image.

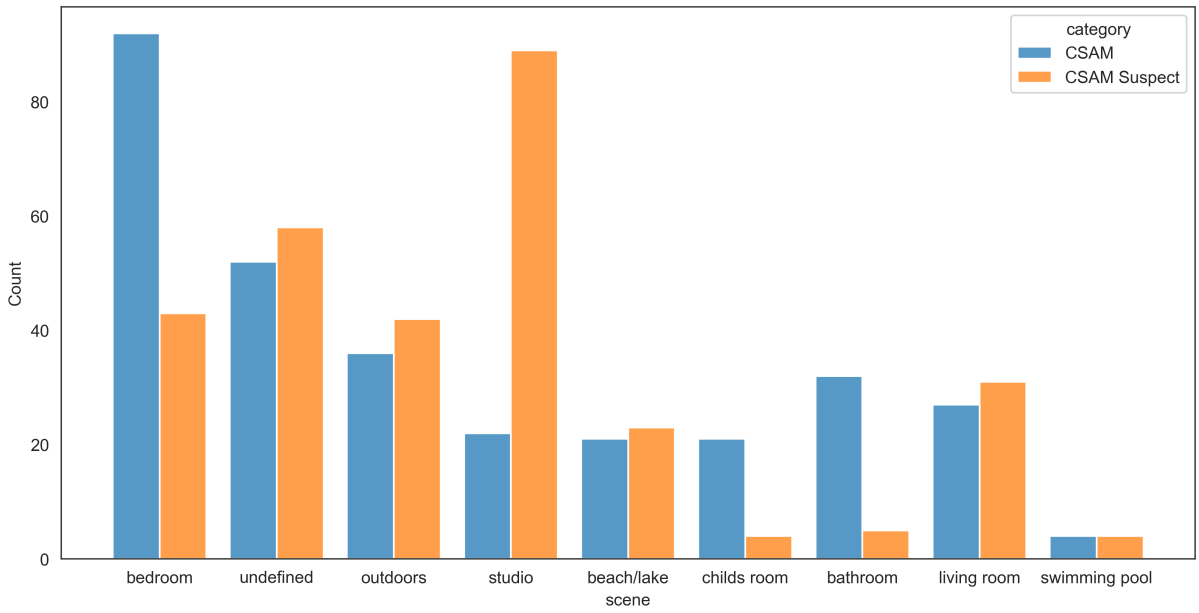


Figure 5.22: Histogram of labeled scenes within CSAM and CSAM Suspect categories.

The model achieved a balanced accuracy of 36.7% among the classes in Places8. This result is lower than what has been seen in the training dataset, and it demonstrates the difficulties in working with this kind of data.

Beyond that, the balanced accuracies for CSAM and CSAM Suspect categories were 40.0% and 34.1%, respectively, showing a balanced performance despite the differences between the two groups.

Finally, to understand the model's weaknesses, we plot the confusion matrix for the classification process in Figure 5.23, in which it is possible to see some clear biases, which were noticed during the manual labeling process.

First, "bathroom" and "swimming pool" seem to be a point of confusion, possibly due to the presence of water in both scenes. Besides, "studio" seems to have been completely confused for "dressing room", an issue which had already been seen in Figure 5.21. In there, studio photography images of people posing had the same misclassification, possibly due to the large portion of the image being occupied by clothes. In that sense, we notice that by exchanging labels between these two classes, the balanced accuracy increases to 45.9%. In the end, "Child's room" was also commonly classified as "bedroom", a problem expected, as there is a level of

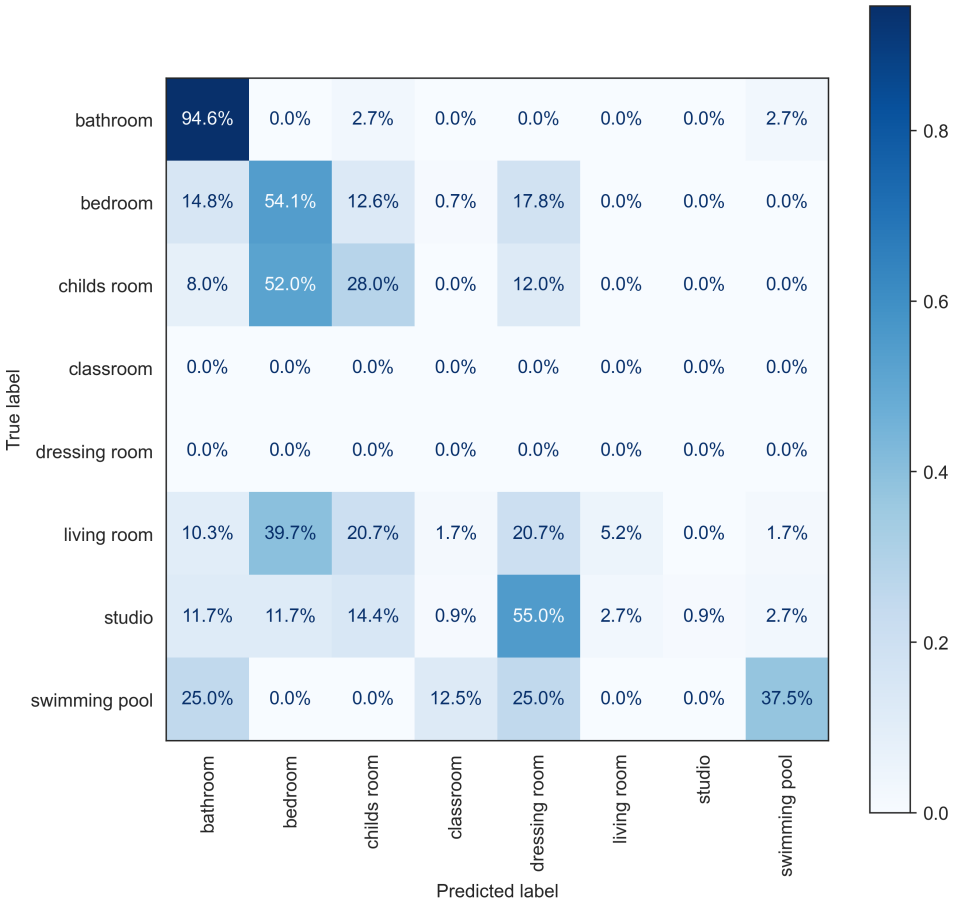


Figure 5.23: Confusion matrix for Places8 scenes classified in the CSAM dataset with values normalized by the number of elements in each class.

ambiguity in many cases.

Ultimately, this test was essential to understand the model's limitations in real-world conditions, with entries that do not fit any of the trained classes and the learned features do not show up with the same distributions. There is a prolonged discussion on the topic in Appendix B.

5.4 Infrastructure and Implementation Details

All experiments were conducted using 6 NVIDIA RTX GPUs A6000 with 48 GiB each. We used PyTorch 1.10[5] as the deep learning framework with the VIsion library for state-of-the-art Self-Supervised Learning (VISSL)[6]. The experiments and model checkpoints were managed with the free education version of Weights & Biases SaaS[7].

Sided with the hardware infrastructure, the deep learning optimizations of Automatic Mixed Precision and Activation Checkpointing were indispensable for training multiple models with multi-million image datasets for many epochs. The objective of all optimizations was to increase the convergence rate, i.e., the time until an experiment reaches the same level of perfor-

[5] https://pytorch.org
[6] https://vissl.ai
[7] https://wandb.ai

mance without it. In that sense, we could get around 1.5 hours per epoch for model pretraining on 8 million images and 3 minutes per epoch for model finetuning on 300,000 images, using a batch size of 1024 images per GPU.

The CSAM test was executed in a GPU workstation owned by a Federal Police Agent, and inference ran on 46,006 images within 30 minutes.

Chapter 6

Conclusions

This Master's thesis applied deep learning tools to tackle scene recognition in both general and specific scenarios. We applied self-supervision to indoor classification to help with CSAM triage and recognition, showing that SSL can leverage scene images to perform better than supervised learning.

Furthermore, we compare SSL performance in multiple conditions, using real and synthetic images, common and sensitive data. The inference in real CSAM is a vital result, bringing the real distribution of scenes in this data and how a model trained with data from the Places dataset [134] performs on it.

Our experiments show that combining publicly available self-supervision models with large scene recognition datasets for a pretext task helps indoor classification, even on the scenes prevalent in CSAM. However, sensitive material apparently shows unique features, which these datasets do not commonly provide.

In the end, we compare various SSL techniques in one specific task and understand the best scenario for its solution. From our perspective, it should serve as a reference for other endeavors in both scene classification and CSAM recognition.

6.1 Answers to Research Questions

The research questions were designed to guide the research process and define the experiment set. Following the experiment results, we answer each question accordingly.

Q1. **Does self-supervised outperform supervised learning for indoor classification?** Yes. Within Places8, our target indoors task, we can assess with over 98% confidence that SSL outperforms supervised methods with an average improvement of 2.2 percentage points in balanced accuracy (Figure 5.19).

Q2. **Can we boost self-supervised target task performance by adding a pretraining task that uses synthetic images and their segmentation maps?** No, not with synthetic images. The usage of synthetic images in the pretext task did not improve the performance of the final model in comparison with the pretext task that only used real images. We hypothesize that using synthetic masks (e.g., depth, segmentation, optical flux, Lambertian) the same way we use real or photorealistic images in

contrastive learning methods impacted convergence and required the model to train longer to reach the same level of performance without it (Figure 5.18). All in all, using a custom-tailored pretext task is welcome when using self-supervision, but more data does not always translate into better performance.

Q3. **Are popular scene recognition datasets representative enough of CSAM environments that a scene classification model built from them can be used on such sensitive data?** No, not for all classes. Based on the analyzed CSAM, we can say some classes from Places8 have matching features while some do not. In particular, "bathroom" and "swimming pool" seem representative enough in both datasets, but Places8 "studio" had almost no match in CSAM and was mostly taken as "dressing room", which was actually absent from the CSAM data. Furthermore, the semantic ambiguity (explained in Chapter 3) among "bedroom", "child's room" and "living room" seems to be too hard for the model to distinguish and seemingly made it look for inconsistent features during classification, e.g., the presence of children or toys for child's room, bed or pillow for bedroom, sofa or coffee table for living room. The results obtained are useful and should serve as a starting point for further investigations on the topic.

6.2 Challenges and Future Work

The level of performance we reached in our target task was optimal under the conditions we worked, but far better results can be achieved if some constraints are either changed or removed.

First, the current solution is not particularly useful for CSAM recognition and triage in its current state, classifying at most two scenes with ideal performance. Given the diversity of features unique to this kind of data, a more robust target task must also be more general: it must be able to classify not only indoor scenes but possibly objects, people, body parts and some outdoor scenes as well. Also, feature fusion with the scene model could improve current CSAM classifiers and improve overall representations.

Additionally, all experiments used ResNet-50, which is fairly popular in SSL research, but multiple other architectures have gained much research interest after their sustained accomplishments on various tasks, such as the larger RegNets [42, 101], and Visual Transformers [18, 27, 33]. Also, ResNet variations have been tailored for scene recognition and can perform better than its general version [129]. It might be possible to adapt the SSL pipeline to these neural networks, even reusing the pretrained weights, to reach improved models without complex changes to the process.

Moreover, we believe the synthetic data was not used properly when setting the pretext tasks, and any synthetic masks (e.g., depth, segmentation, optical flux, Lambertian) should not be treated the same way as real or photorealistic images by the contrastive mechanism. Either a different usage of the synthetic data or an adapted loss should be used. For example, the Indoors.all dataset could be restricted only to photorealistic images, thus reducing its size and possibly converging faster and to better representations. Another viable possibility would be to use the original labels from the synthetic images in a downstream task since some synthetic datasets have the scene labeled.

On a different approach, we tried to provide a comparison using four different SSL models and extensive hyperparametrization, SimCLR [24], SwAV [17], BarlowTwins [128], and SupCon [65], which limited our ability to pretrain for longer epochs. Thus, as Section 5.3.4 discusses, it might be useful to pretrain the models longer, even without changes to the original datasets.

Taking this into account, the largest challenge with SSL pretext tasks is the high cost of memory and compute power required. The usage of the optimization strategies presented in Section 2.3 was essential to the success of the work, and other techniques, such as microbatching [56], a.k.a gradient accumulation, pruning [92], and sharded training [93, 102] could be added to this process for larger convergence rates.

Finally, CSAM is a challenging research topic on both experimental and psychological levels. However, it is an essential subject that society must face and find smart solutions to stop its spread, speeding up police work and simultaneously helping to keep more children safe.

Bibliography

[1] Mixed precision: Tensorflow core. URL `https://www.tensorflow.org/guide/mixed_precision`.

[2] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, M. Gates, T. Grützmacher, N. J. Higham, S. Li et al. A survey of numerical methods utilizing mixed precision arithmetic. *arXiv preprint arXiv:2007.06674*, 2020.

[3] M. W. Al-Nabki, E. Fidalgo, R. A. Vasco-Carofilis, F. Janez-Martino and J. Velasco-Mata. Evaluating performance of an adult pornography classifier for child sexual abuse detection. *arXiv preprint arXiv:2005.08766*, 2020.

[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.

[5] S. Avila, D. Moreira, M. Perez, D. Moraes, V. Testoni, S. Goldenstein, E. Valle and A. Rocha. Multimodal and real-time method for filtering sensitive media, 2019. US Patent 10,194,203.

[6] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu and M. Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312, 2022.

[7] H. B. Barlow. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01), 1961.

[8] D. Bau, B. Zhou, A. Khosla, A. Oliva and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6541–6549, 2017.

[9] A. Bellet, A. Habrard and M. Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[10] Y. Bengio, Y. Lecun and G. Hinton. Deep learning for ai. *Communications of the ACM*, 64(7):58–65, 2021.

[11] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[12] M. R. Boutell, J. Luo, X. Shen and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[14] S. R. Bulò, L. Porzi and P. Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[15] 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.

[16] M. Caron, P. Bojanowski, A. Joulin and M. Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, pages 132–149, 2018.

[17] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, pages 9912–9924, 2020.

[18] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski and A. Joulin. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision*, pages 9650–9660, 2021.

[19] M. Castrillón-Santana, J. Lorenzo-Navarro, C. M. Travieso-González, D. Freire-Obregón and J. B. Alonso-Hernández. Evaluation of local descriptors and cnns for non-adult detection in visual content. *Pattern Recognition Letters*, 113:10–18, 2018.

[20] 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 *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 721–729, 2020.

[21] L. Chaves, A. Bissoto, E. Valle and S. Avila. An evaluation of self-supervised pre-training for skin-lesion analysis. In *European Conference on Computer Vision*, 2022.

[22] G. Chen, X. Song, H. Zeng and S. Jiang. Scene recognition with prototype-agnostic scene layout. *IEEE Transactions on Image Processing*, 29:5877–5888, 2020.

[23] T. Chen, B. Xu, C. Zhang and C. Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[24] T. Chen, S. Kornblith, M. Norouzi and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

[25] X. Chen and K. He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[26] X. Chen, H. Fan, R. Girshick and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

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

[28] X. Cheng, J. Lu, J. Feng, B. Yuan and J. Zhou. Scene recognition with objectness. *Pattern Recognition*, 74:474–487, 2018.

[29] M. Cimpoi, S. Maji and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015.

[30] E. Cole, X. Yang, K. Wilber, O. Mac Aodha and S. Belongie. When does contrastive visual representation learning work? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14755–14764, 2022.

[31] 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.

[32] T. De Vries, I. Misra, C. Wang and L. Van der Maaten. Does object recognition work for everyone? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 52–59, 2019.

[33] 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. In *International Conference on Learning Representations*, 2021.

[34] T. Durand, N. Thome and M. Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4743–4752, 2016.

[35] R. Epstein. The cortical basis of visual scene processing. *Visual Cognition*, 12(6):954–978, 2005.

[36] L. Ericsson, H. Gouk and T. M. Hospedales. How well do self-supervised models transfer? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021.

[37] 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.

[38] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii and K. Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[39] R. Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[40] Y. Gong, L. Wang, R. Guo and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European Conference on Computer Vision*, pages 392–407, 2014.

[41] S. N. Goodman. Toward evidence-based medical statistics. 1: The p value fallacy. *Annals of Internal Medicine*, 130(12):995–1004, 1999.

[42] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021.

[43] P. Goyal, Q. Duval, I. Seessel, M. Caron, M. Singh, I. Misra, L. Sagun, A. Joulin and P. Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022.

[44] P. Goyal, A. R. Soriano, C. Hazirbas, L. Sagun and N. Usunier. Fairness indicators for systematic assessments of visual feature extractors. In *ACM Conference on Fairness, Accountability, and Transparency*, page 70–88, 2022.

[45] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284, 2020.

[46] E. Guerra and B. G. Westlake. Detecting child sexual abuse images: traits of child sexual exploitation hosting and displaying websites. *Child Abuse & Neglect*, 122:105336, 2021.

[47] M. Hayat, S. H. Khan, M. Bennamoun and S. An. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, 25(10):4829–4841, 2016.

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

[49] K. He, R. Girshick and P. Dollár. Rethinking imagenet pre-training. In *IEEE International Conference on Computer Vision*, pages 4918–4927, 2019.

[50] K. He, H. Fan, Y. Wu, S. Xie and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[51] J. M. Henderson and A. Hollingworth. High-level scene perception. *Annual review of psychology*, 50(1):243–271, 1999.

[52] U. K. Her Majesty's Government. Tackling child sexual abuse strategy. pages 1–90, 2021.

[53] L. Herranz, S. Jiang and X. Li. Scene recognition with cnns: objects, scales and dataset bias. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 571–579, 2016.

[54] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[55] E. Hoffer, I. Hubara and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[56] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in Neural Information Processing Systems*, 32, 2019.

[57] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, page 448–456, 2015.

[58] U. Iqbal, P. Molchanov, T. B. J. Gall and J. Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *European Conference on Computer Vision*, pages 118–134, 2018.

[59] 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.

[60] H. Jahankhani, B. Akhgar, P. Cochrane and M. Dastbaz. *Policing in the Era of AI and Smart Societies*. Springer, 2020.

[61] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

[62] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[63] D. H. Johnson. The insignificance of statistical significance testing. *The journal of wildlife management*, pages 763–772, 1999.

[64] N. Kehtarnavaz. Chapter 11 - lab project examples. In N. Kehtarnavaz, editor, *Real-Time Digital Signal Processing*, pages 267–283. Newnes, Burlington, 2005.

[65] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[66] D. B. Kirk and W. mei W. Hwu. Chapter 6 - numerical considerations. In D. B. Kirk and W. mei W. Hwu, editors, *Programming Massively Parallel Processors (Third Edition)*, pages 131–147. Morgan Kaufmann, third edition edition, 2017.

[67] 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.

[68] S. Kornblith, J. Shlens and Q. V. Le. Do better imagenet models transfer better? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[69] A. Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[70] A. Krizhevsky, I. Sutskever and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[71] J. K. Kruschke. Bayesian estimation supersedes the t test. *Journal of Experimental Psychology: General*, 142(2):573, 2013.

[72] C. Laranjeira, A. Lacerda and E. R. Nascimento. On modeling context from objects with a long short-term memory for indoor scene recognition. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 249–256, 2019.

[73] C. Laranjeira, J. Macedo, S. Avila and J. dos Santos. Seeing without looking: Analysis pipeline for child sexual abuse datasets. In *ACM Conference on Fairness, Accountability, and Transparency*, page 2189–2205. Association for Computing Machinery, 2022.

[74] G. Larsson, M. Maire and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017.

[75] Y. LeCun, Y. Bengio and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[76] 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.

[77] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang and S. Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference*, 2018.

[78] Y. Li, M. Dixit and N. Vasconcelos. Deep scene image classification with the mfafvnet. In *IEEE International Conference on Computer Vision*, pages 5746–5754, 2017.

[79] Z. Li, T.-W. Yu, S. Sang, S. Wang, M. Song, Y. Liu, Y.-Y. Yeh, R. Zhu, N. Gundavarapu, J. Shi, S. Bi, Z. Xu, H.-X. Yu, K. Sunkavalli, M. Hašan, R. Ramamoorthi and M. Chandraker. Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. *arXiv: 2007.12868*, 2021.

[80] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu and M. Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128 (2):261–318, 2020.

[81] S. Liu, G. Tian and Y. Xu. A novel scene classification model combining resnet based transfer learning and data augmentation with a filter. *Neurocomputing*, 338:191–206, 2019.

[82] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37, 2016.

[83] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

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

[85] J. Macedo, F. Costa and J. A. dos Santos. A benchmark methodology for child pornography detection. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 455–462, 2018.

[86] G. Macilotti. Online child pornography: Conceptual issues and law enforcement challenges. In *Handbook of Research on Trends and Issues in Crime Prevention, Rehabilitation, and Victim Support*, pages 226–247. 2020.

[87] S. McCandlish, J. Kaplan, D. Amodei and O. D. Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[88] J. McCormac, A. Handa, S. Leutenegger and A. J. Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *IEEE International Conference on Computer Vision*, pages 2697–2706, 2017.

[89] H. L. Merdian, N. Wilson, J. Thakker, C. Curtis, D. P. Boer et al. "So why did you do it?": Explanations provided by child pornography offenders. *Sexual Offender Treatment*, 8(1):1–19, 2013.

[90] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[91] S. Mittal, H. Gupta and S. Srivastava. A survey on hardware security of dnn models and accelerators. *Journal of Systems Architecture*, 117:102163, 2021.

[92] H. Mostafa and X. Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pages 4646–4655, 2019.

[93] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

[94] A. v. d. Oord, Y. Li and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[95] A. Panchenko, R. Beaufort and C. Fairon. Detection of child sexual abuse media on p2p networks: Normalization and classification of associated filenames. In *LREC Workshop on Language Resources for Public Security Applications*, pages 27–31, 2012.

[96] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[97] C. Peersman, C. Schulze, A. Rashid, M. Brennan and C. Fischer. icop: Automatically identifying new child abuse media in p2p networks. In *IEEE Security and Privacy Workshops*, pages 124–131, 2014.

[98] 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.

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

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

[101] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He and P. Dollár. Designing network design spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

[102] S. Rajbhandari, J. Rasley, O. Ruwase and Y. He. Zero: Memory optimizations toward training trillion parameter models. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2020.

[103] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[104] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

[105] J. Redmon, S. Divvala, R. Girshick and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[106] Z. Ren and Y. J. Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 762–771, 2018.

[107] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb and J. M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision*, 2021.

[108] W. A. G. Rojas, S. Diamos, K. R. Kini, D. Kanter, V. J. Reddi and C. Coleman. The dollar street dataset: Images representing the geographic and socioeconomic diversity of the world. *arXiv preprint arXiv:2201.08371*, 2022.

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

[110] N. Sae-Bae, X. Sun, H. T. Sencar and N. D. Memon. Towards automatic detection of child pornography. In *IEEE International Conference on Image Processing*, pages 5332–5336, 2014.

[111] N. Sharad Sohoni, C. R. Aberger, M. Leszczynski, J. Zhang and C. Ré. Low-memory neural network training: A technical report. *arXiv preprint arXiv:1904.10631*, 2019.

[112] D.-Y. She and K. Xu. Contrastive self-supervised representation learning using synthetic data. *International Journal of Automation and Computing*, 18(4):556–567, 2021.

[113] A. Shupo, M. V. Martin, L. Rueda, A. Bulkan, Y. Chen and P. C. Hung. Toward efficient detection of child pornography in the network infrastructure. *IADIS International Journal on Computer Science and Information Systems*, 1(2):15–31, 2006.

[114] S. Song, S. P. Lichtenberg and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015.

[115] A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges and J. Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *European Conference on Computer Vision*, pages 211–228, 2020.

[116] A. Straw, T. Wiecki, C. Fonnesbeck and S. Andrés. Bayesian estimation supersedes the t-test. In P. Team, editor, *PyMC examples*. doi: 10.5281/zenodo.5654871.

[117] A. Tabone, K. Camilleri, A. Bonnici, S. Cristina, R. Farrugia and M. Borg. Pornographic content classification using deep-learning. In *ACM Symposium on Document Engineering*, pages 1–10, 2021.

[118] Y. Tian, D. Krishnan and P. Isola. Contrastive multiview coding. In *European Conference Computer Vision*, pages 776–794, 2020.

[119] L. Treszkai. Model version history. URL `https://best.readthedocs.io/en/latest/model_history.html`.

[120] 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.

[121] R. Wu, B. Wang, W. Wang and Y. Yu. Harvesting discriminative meta objects with deep cnn features for scene classification. In *IEEE International Conference on Computer Vision*, pages 1287–1295, 2015.

[122] Z. Wu, Y. Xiong, S. X. Yu and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[123] L. Xie, F. Lee, L. Liu, K. Kotani and Q. Chen. Scene recognition: A comprehensive survey. *Pattern Recognition*, 102:107205, 2020.

[124] S. Yang and D. Ramanan. Multi-scale recognition with dag-cnns. In *IEEE International Conference on Computer Vision*, pages 1215–1223, 2015.

[125] E. Yiallourou, R. Demetriou and A. Lanitis. On the detection of images containing child-pornographic material. In *International Conference on Telecommunications*, pages 1–5, 2017.

[126] Y. You, I. Gitman and B. Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[127] 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*, 2020.

[128] J. Zbontar, L. Jing, I. Misra, Y. LeCun and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320, 2021.

[129] D. Zeng, M. Liao, M. Tavakolian, Y. Guo, B. Zhou, D. Hu, M. Pietikäinen and L. Liu. Deep learning for scene classification: A survey. *arXiv preprint arXiv:2101.10531*, 2021.

[130] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi and A. Agrawal. Context encoding for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[131] Z. Zhao and M. Larson. From volcano to toyshop: Adaptive discriminative region discovery for scene recognition. In *ACM International Conference on Multimedia*, pages 1760–1768, 2018.

[132] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba and A. Oliva. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, 27, 2014.

[133] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba. Learning deep features for discriminative localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[134] 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.

[135] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk and Q. V. Le. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems*, pages 3833–3845, 2020.

[136] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang and Y. Chen. Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Transactions on Image Processing*, 25(7):2983–2996, 2016.

Appendix A

Datasheet for Litmus Test Dataset

Simultaneously important as obtaining data is documenting its process of acquisition and purposes. When working with data-driven technologies, transparency is crucial to prevent undesired, possibly harmful results and allow for constructive critique. With that in mind, we present the datasheet for our Litmus Test dataset, as proposed by Gebru et al. [38].

Motivation

1. **For what purpose was the dataset created?** We attempt to build a small yet diverse dataset to check if the produced model performance holds outside of the controlled nature of Places8.

2. **Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?** The author of this Master's thesis created the dataset on behalf of the Institute of Computing, University of Campinas.

3. **Who funded the creation of the dataset?** No funding.

Composition

1. **What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** Photos of places and environments.

2. **How many instances are there in total (of each type, if appropriate)?** There are 10 images per category, with 8 categories totaling 80 images.

3. **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** It is a sample of images: 5 images were selected from Google images and another 5 from Dollar Street, each from a different household (Dollar Street contains 289 households). When Dollar Street did not contain the specified class, the 5 images were taken from Bing images for variety.

4. **What data does each instance consist of?** JPEG images

5. **Is there a label or target associated with each instance?** Yes. Each image contains one of 8 possible labels: bathroom, bedroom, child's room, classroom, dressing room, living room, studio, and swimming pool.
6. **Is any information missing from individual instances?** No.
7. **Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?** There is no direct relation besides the label, but the source of each image (Google/Bing images or Dollar Street) are distinguished in the metadata file accompanying the dataset.
8. **Are there recommended data splits (e.g., training, development/validation, testing)?** No, the dataset is created for testing only.
9. **Are there any errors, sources of noise, or redundancies in the dataset?** No.
10. **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** The dataset is self-contained, i.e., it contains all images, but some of the original links to the images are provided in the metadata file.
11. **Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non-public communications)?** No.
12. **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** No.
13. **Does the dataset relate to people?** The dataset comprises indoor environments of people's homes, and some images contain people.
14. **Does the dataset identify any subpopulations (e.g., by age, gender)?** The Dollar Street original images list income and country.
15. **Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?** Yes, people are found in some images.
16. **Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals race or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** No.

Collection

1. **How was the data associated with each instance acquired?** The dataset is a combination of images collected online. Some images were found on Google/Bing using the class name as a search keyword, while others were taken from the associated label in the Dollar Street dataset.

2. **What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?** The images were collected manually to fetch diverse enough environments that would be hard for the model to classify appropriately.

3. **If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?** There is no specific sampling process. We captured 5 images from the three possible sources with manual human curation.

4. **Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?** The author of this Master's thesis.

5. **Over what timeframe was the data collected?** All dataset was collected on July 2nd, 2022.

6. **Were any ethical review processes conducted (e.g., by an institutional review board)?** No.

7. **Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?** The data was obtained from third parties: Google/Bing images and the Dollar Street dataset.

8. **Were the individuals in question notified about the data collection?** All Dollar Street images are taken with the consent of the owners of the place.

9. **Did the individuals in question consent to the collection and use of their data?** All Dollar Street images are taken with the consent of the owners of the place.

10. **If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?** Not for this dataset, but the third-party providers enable this resource.

11. **Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?** No.

Preprocessing

1. **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** Yes, all images are JPEG compressed. No other preprocessing is done.

2. **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** Raw images are not provided.

3. **Is the software used to preprocess/clean/label the instances available?** No.

Uses

1. **Has the dataset been used for any tasks already?** Yes. This Master's thesis used this dataset as an inference test for deep learning models.
2. **Is there a repository that links to any or all papers or systems that use the dataset?** No.
3. **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** No.
4. **Are there tasks for which the dataset should not be used?** No.

Distribution

1. **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** Yes, a public repository for keeping the files is shared.
2. **How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?** GitHub repository.
3. **When will the dataset be distributed?** When this Master's thesis is concluded and publicly released.
4. **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** No. All images are originally free to use, edit and share.
5. **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** No.
6. **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No.

Maintenance

1. **Who will be supporting/hosting/maintaining the dataset?** The author of this Master's thesis.
2. **How can the owner/curator/manager of the dataset be contacted (e.g., email address)** GitHub issues open in the repository are the preferred method, but the email address is also shared.
3. **Is there an erratum?** No, but a changelog can be appended to the repository and serve as a reference for version management.

4. **Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?** Yes, until this Master's thesis is publicly released.

5. **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?** No.

6. **Will older versions of the dataset continue to be supported/hosted/maintained?** Yes, older versions can be found in the repository as well.

7. **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** One can open a GitHub issue, pull a request, or fork the repository to make changes.

Appendix B

Considerations on CSAM Data

The CSAM test enabled a real-world evaluation of our process, but the created model was limited to 8 classes, whereas the annotated data had more possible scenes, including ‘undefined‘, which the model does not expect.

Figure B.1 complements the analysis of the model performance, by including the classes of “beach/lake”, “outdoors” and “undefined” in the confusion matrix. Seemingly, the presence of water or shades of blue favors the classes “bathroom” and “swimming pool”, which explain why “beach/lake” (water) or “outdoors” (sky) were labeled as such. On the other hand, “undefined” had distributed results, not focusing on one specific scene.

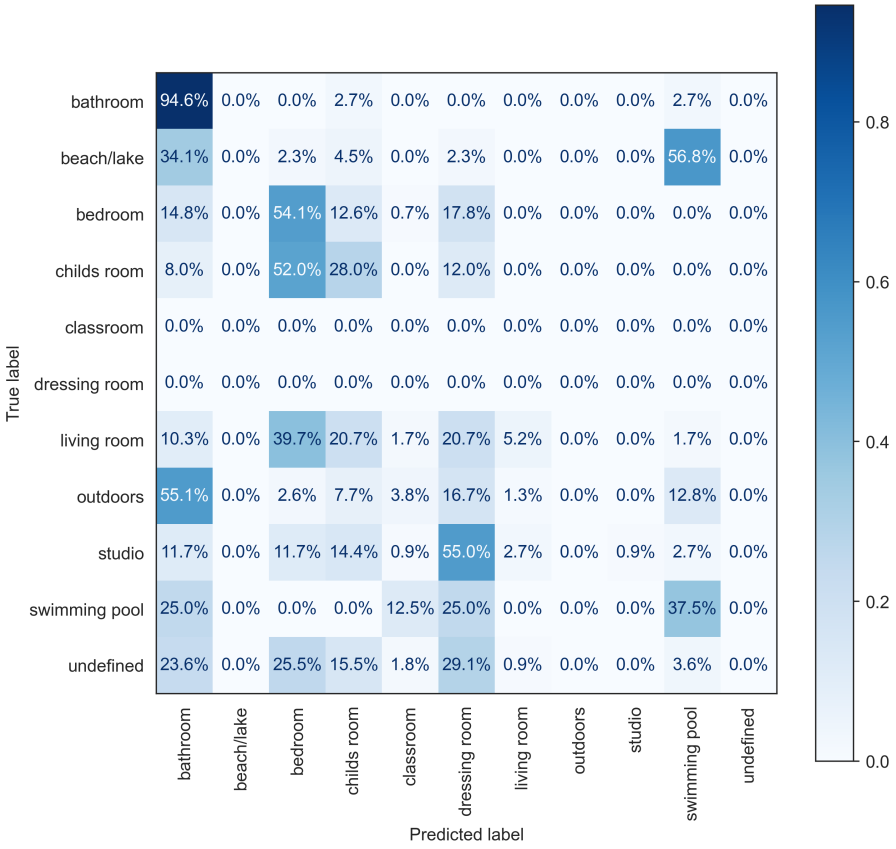


Figure B.1: Confusion matrix for all scenes classified in the CSAM dataset with values normalized by the number of elements in each class.

Beyond classifying scenes, this experiment provides the chance to analyze patterns in CSAM from a different perspective. Looking at all 46,006 images in the dataset, we have other categories besides “CSAM” and “CSAM Suspect”, which are possible images found in the apprehended computers. Namely, the other possible categories are “Cartoon”, “Money Bill”, “Digitalization”, “Document”, “Violence”, “People”, “Pornography” and “Others” and their distribution is represented on the histogram in Figure B.2.

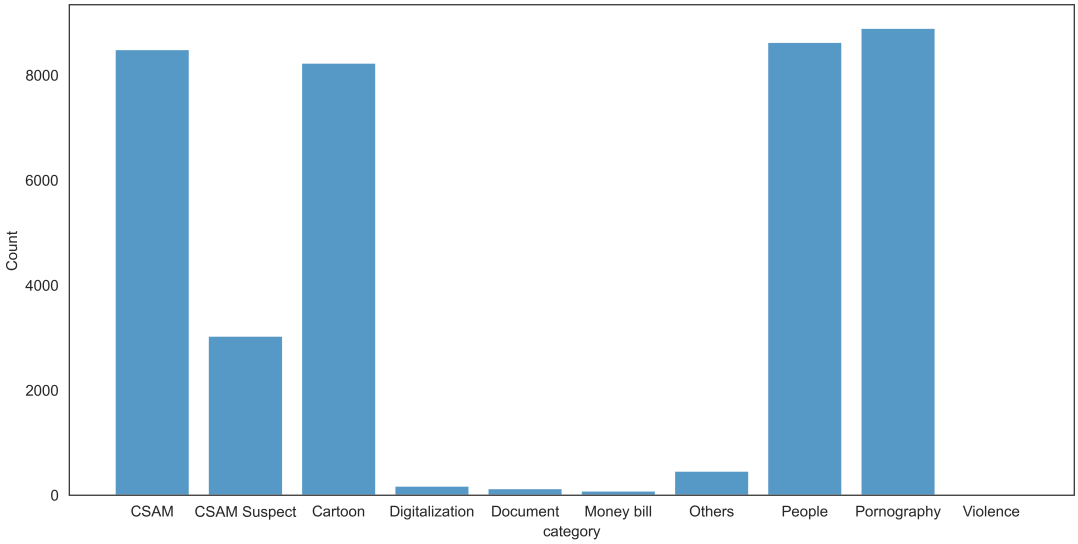


Figure B.2: Histogram of CSAM dataset categories.

Moreover, Figure B.3 shows the distribution of each classified scene in each dataset category. It is visible that some categories seem to have unique characteristics, which can match the features the scene classifier is looking for, e.g., most “child’s room” seems to be concentrated in the “cartoon” bucket, which is a known bias of Places365, that includes toy and cartoon advertisements in the “child’s room” class [134].

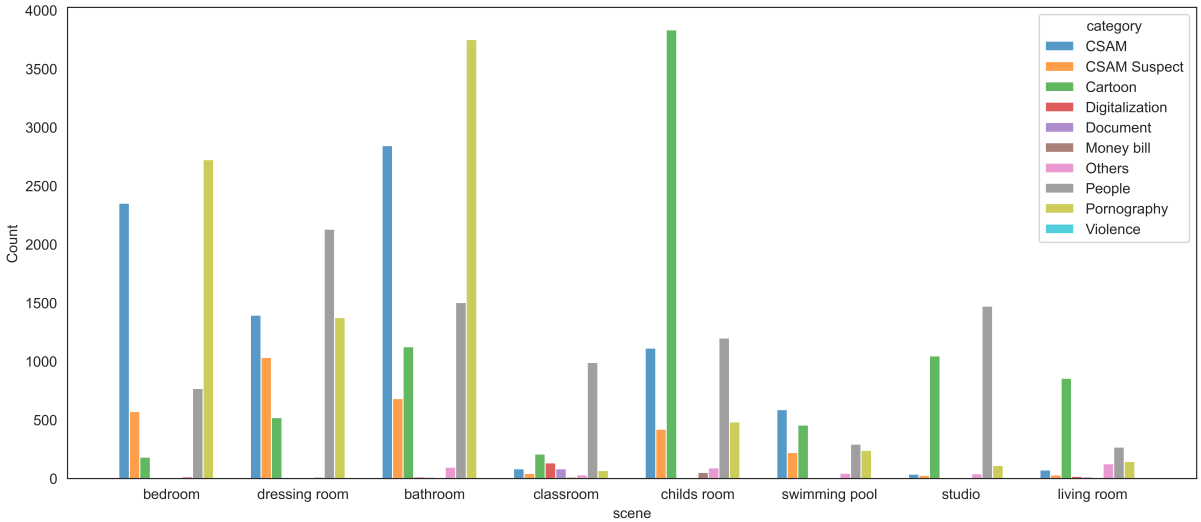


Figure B.3: Histogram of labeled scenes within all of the dataset categories.

Even considering the errors and bias, we can take the different categories of the dataset and correlate them with each other through the distribution of scenes. Figure B.4 shows the

correlation matrix following this process and we highlight the findings that “Document” is similar to “Digitalization”, that “Money bill” follows “Cartoon”, but in special that “Pornography” and “CSAM” have quite high correlation. Even though there are no scenes in “Document” or “Digitalization” for instance, the similarity of the images in these groups from the model’s perspective led it to classify it with equally similar distributions of scenes.

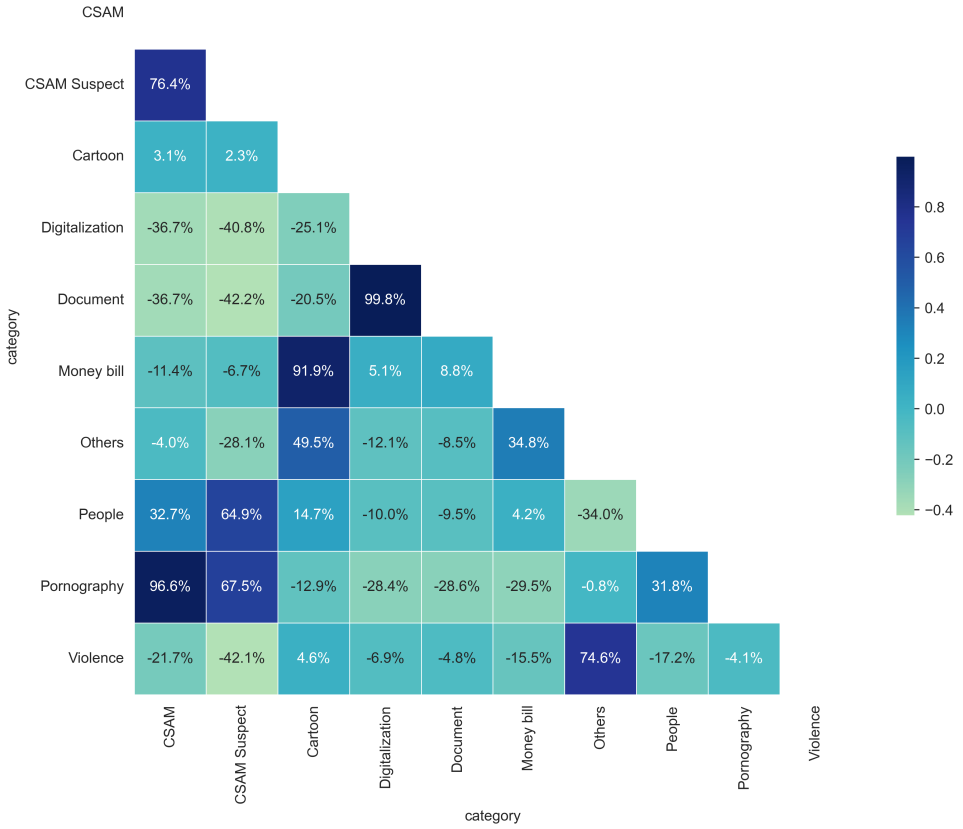


Figure B.4: Correlation matrix of different CSAM categories based on the distribution of classified scenes. Pearson’s linear coefficient is used as a measurement of correlation.

We emphasize the importance of this last remark: whereas the other categories do not represent places, “Pornography” and “CSAM” to have the same distribution of scenes is an important connection between the two groups that, in principle, could not be true.

In other words, these two groups differentiate themselves in legal terms, but there is likely a high similarity in how they are depicted. These two classes can be hard to differ when looking only through the lens of scene recognition, an essential consideration if applied in the real world. Nevertheless, CSAM and CSAM Suspect are different in meaning and in the distribution of their scenes.

These results show how this kind of triage is not indirectly helpful but fits into the problem of child sexual abuse recognition. To put it simply, the action correlates to the place it happens.