Visual Recognition of Human Rights Violations

Kalliatakis Grigorios

School of Computer Science and Electronic Engineering
University of Essex

A thesis submitted for the degree of
Doctor of Philosophy

November 2019

Abstract

This thesis is concerned with the automation of human rights violation recognition in images. Solving this problem is extremely beneficial to human rights organisations and investigators, who are often interested in identifying and documenting potential violations of human rights within images. It will allow them to avoid the overwhelming task of analysing large volumes of images manually. However, visual recognition of human rights violations is challenging and previously unattempted. Through the use of computer vision, the notion of *visual recognition of human rights violations* is forged in this thesis, whilst this area is addressed by strongly considering the constraints related to the usability and flexibility of a real practice. Firstly, image datasets of human rights violations which are suitable for training and testing modern visual representations, such as convolutional neural networks (CNNs) are introduced for the first time ever. Secondly, we develop and apply transfer learning models specific to the human rights violation recognition problem. Various fusion methods are proposed for performing an equivalence and complementarity analysis of object-centric and scene-centric deep image representations for the task of human rights violation recognition. Additionally, a web demo for predicting human rights violations that may be used directly by human rights advocates and analysts is developed. Next, the problem of recognising displaced people from still images is considered. To solve this, a novel mechanism centred around the level of control each person feels of the situation is developed. By leveraging this mechanism, typical image classification turns into a uniform framework that infers potential displaced people from images. Finally, a human-centric approach for recognising rich information about two emotional states is proposed. The derived *global emotional traits* are harnessed alongside a data-driven CNN classifier to efficiently infer two of the most widespread modern abuses against human rights, child labour and displaced populations.

Declaration

This thesis is submitted to the School of Computer Science and Electronic Engineering, University of Essex, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Kalliatakis Grigorios
November 2019

To Christina

For her advice, her patience, and her faith,
because she always understood.

Acknowledgements

I am extremely grateful to my wonderful group of supervisors, Klaus D. McDonald-Maier and Shoaib Ehsan, whose constant guidance, enthusiasm and inspiration enabled the completion of this thesis. Over the course of this PhD I was very lucky to interact with Aleš Leonardis and Maria Fasli. The work I have done in collaboration with them forms the essence of this thesis. A big thank you to the Embedded and Intelligent Systems (EIS@Essex) Laboratory members for making the lab an excellent environment. I am also grateful to Alexandros Stergiou, not only for reviewing this thesis, but also for the useful discussions and encouragement during my PhD study. And I would not be in the field of computer vision without the support of Georgios Triantafyllidis. Finally, all my appreciation to my fiancée and family for the many years of unconditional support.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ARCADE	Artillery Crater Analysis and Detection Engine
CAM	Class Activation Mapping
CIFAR	Canadian Institute for Advanced Research
CNN	Convolutional Neural Network
DNN	Deep Neural Network
E-LAMP	Event Labeling through Analytic Media Processing
FC	Fully Connected
GET-AID	Global Emotional Traits for Abuse Identification
GET	Global Emotional Traits
GPU	Graphics Processing Unit
Grad-CAM	Gradient-weighted Class Activation Mapping
HOG	Histogram of Oriented Gradients
HRA	Human Rights Archive Dataset
HRUN	Human Rights Understanding Dataset
HRVR	Human Rights Violation Recognition

IDP	Internally Displaced People
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
mAP	Mean Average Precision
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
MTurk	Amazon Mechanical Turk
NGO	Non-Governmental Organization
NIN	Network in Network
OHCHR	Office of the High Commissioner for Human Rights
PCA	Principal Component Analysis
R-CNN	Region with CNN Features
ReLU	Rectified Linear Unit
RPN	Region Proposal Networks
SGD	Stochastic Gradient Descent
SIFT	Scale Invariant Feature Transform
SVHN	Street View House Numbers
SVM	Support Vector Machine
UNHCR	United Nations High Commissioner for Refugees
VAD	Valence Arousal Dominance
VLAD	Vector of Locally Aggregated Descriptors

Chapter 1

Introduction

Computer vision is undergoing a period of massive expansion. This is not because computers have achieved human-like perception, but because of advances in *large scale deep learning*, where computers learn from massive image databases how to classify new data. At the cutting edge are the neural networks that have learned to recognise objects or optical characters, a small core of computer vision goals aimed at replicating human abilities. One activity that currently seems distant from computer vision is human rights advocacy where little empirical research has documented the influence images can have in bringing human rights topics to life in a way that mere description and texts cannot. This chapter offers reflections on some of the key challenges of visual recognition in illuminating gross violations of human rights. The major contributions made by this thesis in an attempt to bridge these research gaps are also highlighted. A snapshot of each chapter is presented to illustrate the structure of the thesis. Finally, publications that were made during the course of this research are listed at the end of the chapter.

1.1 Objectives and Motivations

Violations of human rights have been unfolding during the entire human history, while nowadays they appear in many different forms around the world. In the era of social media and big data, publicly available footage is becoming an increasingly important aspect of conflict monitoring and the documentation of war crimes and human rights abuse [75]. Human rights organisations, advocates, journalists, international institutions, and ordinary people find themselves drowned in massive amounts of visual testimony of suffering and misconduct. The ubiquity of such visual data may deluge those accountable for analysing and preserving them. Currently, the workflow followed by humanitarian and human rights professionals is to seek out, verify and edit the most disturbing and traumatic raw images captured by consumer cameras 'in the wild' and posted online. This involves manual sifting through massive volumes of eyewitness media images and videos and looking at, or watching footage over and over again in order to extract useful information [50, 89]. Such analysis most of the time is utterly expensive (when people must be paid to do the work), time consuming, and emotionally traumatic [15]. Furthermore, the number of researches or volunteers who are capable of carrying out such work can be limited by language skills, geographic awareness, and cultural knowledge.

Visual recognition of events where human rights are potentially being violated plays a crucial role in human rights advocacy and accountability efforts. While, manual processing is sufficient for small-scale visual data, the circulation of human-rights-related content has largely outclassed the ability of researchers to keep pace. Hence, automatic perception of human rights violations will enable researchers to discover content that may otherwise be concealed by massive volume of images and videos. These automated systems are not producing evidence, but are instead narrowing down the amount of material that must be examined by human analysts to improve their reporting, operations, storytelling, investigations, prosecutions and advocacy.

The research into computer vision and machine learning have seen tremendous progress in recent years, due to the advances in deep learning. However, the vast majority of research conducted covers broad areas such as object recognition, image classification and semantic segmentation; progress related to visual recognition of human rights violations has been non existent, somewhat due to the insufficient availability of training data. The objective of this thesis is to bridge this gap and help improve the efficiency and effectiveness of human rights practitioners who analyse imagery as a significant dimension of their work.

In this thesis, we aim to automate the process of recognising potential human rights violations from images which would entail in developing a purpose-built *human rights technology* well suited for sifting through large-scale image collections and outputting action-provoking samples, specifically designed for modern human rights practice. This automated visual

(a)

(b)

Figure 1.1: Ambiguities in visual recognition of human rights violations.

recognition system would aid advocates in dealing with a large amount of visual evidence. An additional asset of this system is considered the standardised definition of human rights violations; images of human rights violations have been gathered by verified sources. Such an automated system could be deployed by different organisations and advocates around the world, thus eliminating a significant concern associated with human rights technology [78]; processing large imagery collections tends to be limited to institutions with large staffs or access to expensive, technologically advanced tools and techniques. Specific contributions are described in Section 1.3.

1.2 Key Challenges

Ambiguities in visual recognition of human rights violations. Visual recognition of human rights violations is considered to be a particularly challenging task, and it is understood that even experienced analysts find it difficult to tell the entire story of an event or present an issue 'as it really is' even with high volume of visual data [3]. Ambiguity of static images can often lead to different interpretations. For example, Figure 1.1 can be interpreted in different ways: (a) can swiftly be interpreted as a boy wandering in countryside carrying some personal possessions. At the same time, one might put that in different context and claim that the young boy is carrying those belongings because he is actually forced to sell them (*child labour*). Similarly, (b) might be interpreted as a group of children commonly posing for a picture, while someone else might interpret it as members of displaced populations that were recorded in a refugee camp. This presents two challenges: the first is *how to gather structured visual knowledge*, and the second is *how to break down such complex dependencies into simpler tasks*.

Variability in places and events. Automated recognition of potential human rights violations is challenging because of high intra- and inter-class variability. For example, regarding *refugee rights*, there is a vast number of events behind this variability, including: (i) *asylum seekers*,

(a) (b) (c)

Figure 1.2: Intra-class variability for violations against refugee rights. Source: Human Rights Watch [83].

such as peoples' access to asylum is blocked, and depriving asylum seekers of rights to fair hearings of their refugee claims; (ii) *internally displaced people* including the forcible return of people to places where their lives or freedom would be threatened; (iii) *migrants*, who are treated without dignity and regard for the basic human rights. An example of intra-class variability for violations against refugee rights is depicted in Figure 1.2: (a) a wooden boat carrying 29 people, mainly Syrians, just before their rescue and transfer to the Aquarius. (b) asylum seekers and migrants, mostly from Syria and Iraq, try to warm themselves after a freezing night near the Greek border with Macedonia. (c) a young Egyptian man in the main room at the Pozzallo Hotspot, Italy.

Lack of training data. Many notable applications of computer vision and machine learning in recent years have been in the area of supervised learning, thanks to the availability of large datasets such as ImageNet [12] and Places [119]. However, such readily available image datasets do not exist in the field of human rights. This raises two challenges: (i) how to generate a collection of labelled data from existing sources (*e.g.* from social media images, videos or non-governmental organizations (NGO)) with minimum human intervention; (ii) how to verify those data.

Incorporate contextual reasoning. In computer vision, context is viewed as a way of structuring knowledge and modelling its usage in problem solving tasks. It is generally accepted that the surroundings of an object may have a huge influence on, and in some cases, may be necessary for, visual recognition of an object [67]. The human ability of inferring information about a scene that is then useful for interpreting other parts of the image, is a remarkable trait of our visual system, one that is required to be modelled by computer vision techniques in order to advance automated recognition of potential human rights violations.

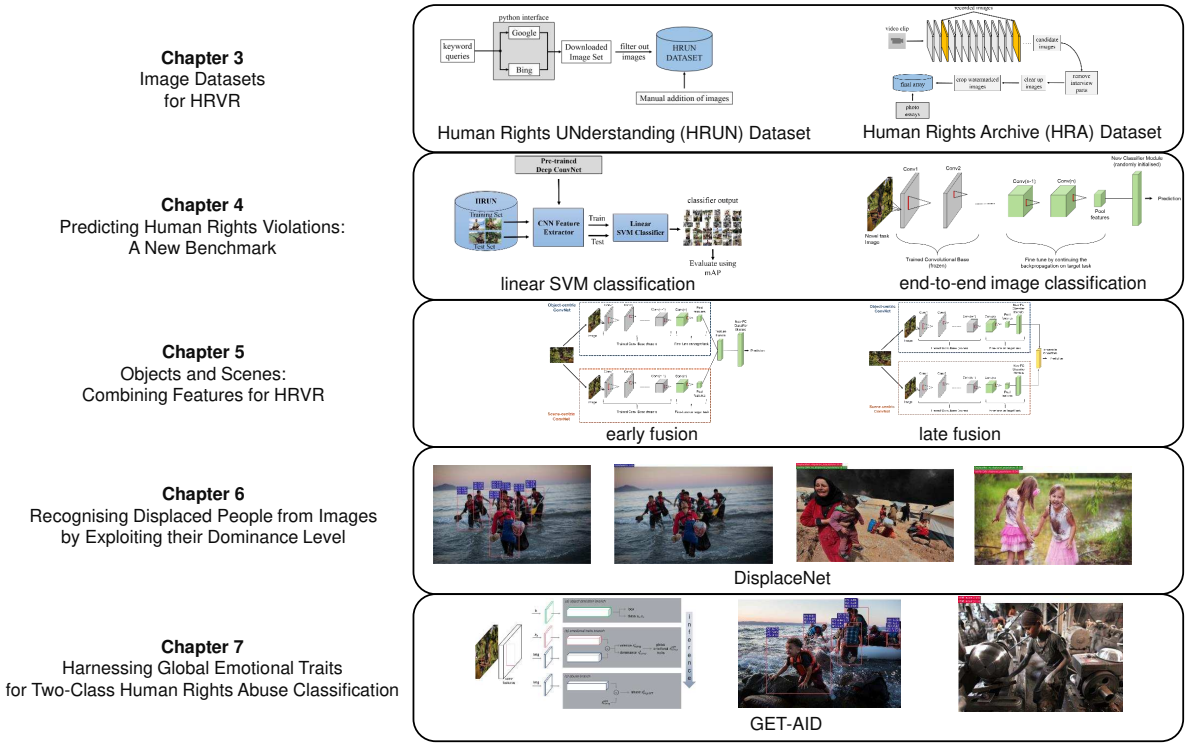


Figure 1.3: An overview of the main contributions presented in this thesis. Firstly, we introduce the first ever image datasets in the context of human rights violations. Then, using various transfer learning methods we introduce the first benchmark for visual recognition of human rights violations. After that, we combine features from object-centric and scene-centric convolutional neural networks for predicting human rights violations. In Chapter 6 we move to recognising displaced people by exploiting their dominance level with an integrated model called *DisplaceNet*. Lastly, global emotional traits–a powerful approach capable of characterising an image based on the emotional states of all people in a scene–is introduced in the recognition pipeline with a clean end-to-end system, called *GET-AID*.

1.3 Contributions

The main contributions of this thesis are fivefold and are summarised in the following list. Figure 1.3 may also be consulted for a condensed overview.

- We construct the first-ever image datasets containing instances of human rights violations, captured in real world situations and surroundings. These datasets signify an attempt to establish a comprehensive set of benchmarks for visual recognition in the context of human rights violations within the vision community for the first time ever. They also lay the foundation for training deep visual representations with the objective to expose human rights violations over large-scale data that may otherwise be impossible.

- Although the latest generation of Convolutional Neural Networks (CNNs) have achieved impressive results in challenging benchmarks on image recognition and object detection, it remains still unclear whether those image representations can be utilised with the same efficiency for complex tasks. With our novel image datasets we tackle the unattempted task of predicting human rights violations from still images, while we quantitatively and qualitatively show by what means deep image representations can be used for this task. We conduct a rigorous evaluation of these image representations exploring different deep architectures and comparing them on a common ground, identifying and disclosing important implementation details. We also propose a two-phase transfer learning framework that can be tailored in an end-to-end image classification system, resulting in the first benchmark for visual recognition of human rights violations.

- Predicting potential human right violations principally consists of more basic tasks, such as object and scene recognition. In this context, we investigate whether features emerging from models that have been trained on objects (object-centric CNNs) and features emerging from models that have been trained on scenes (scene-centric CNNs) can be effectively combined for recognising potential human rights violations. We analyse and empirically clarify their complementarity, conducting a large set of experiments. We also propose various mechanisms for different early and late fusion strategies. We found that recognition of human rights violations poses a challenge at a higher level for the well studied representation learning methods.

- We develop *DisplaceNet*, a novel framework for recognising displaced people from images. Our hypothesis is that the control level of a situation by the person, ranging from *submissive / non-control* to *dominant / in-control*, is a powerful cue that can help our baseline models make a distinction between displaced people and non-displaced people. As a result, we propose a novel method to delineate an image on the basis of all people's dominance level. The apparent dominance level of each person is predicted by jointly processing the window of the person and the whole image. We apply this method to evaluate our framework using a specially adapted image dataset. Results show a significant improvement over our baseline system.

- Our findings indicate that emotional states of people can be closely related with certain abuses against human rights. This discovery served as the basis of our next method called *global emotional traits (GET).* We define global emotional traits as two different scores– they derive from arousal and dominance emotional states–that characterise an entire image. We integrate this method with a data-driven CNN classifier and we introduce an end-to-end system called *GET-AID*. This system results in a significant improvement

over fine-tuned models for two of the most reported modern human rights violations, *child labour* and *displaced people*.

1.4 Thesis Outline

Literature Review. (Chapter 2) We review the existing relevant literature that serves as background for the research conducted in this thesis. We start by examining the role of images in the context of human rights and humanitarian communication, how and why visual knowledge shapes human rights technology, and a variety of studies related to human rights violations. Then we review the literature related to image representation learning which is fundamental in visual recognition, and how it has advanced throughout the years. Finally, we provide a literature review for large-scale image datasets, object detectors, emotion recognition systems, and transfer learning.

Datasets & Verification. (Chapter 3) We introduce the first-ever image datasets suitable for training and testing modern deep CNNs for the task of *human rights violation recognition (HRVR)*. Also, we develop a multi-stage pipeline for fully verified large-scale image collection from NGO repositories. With this, we generate a dataset with over 3K instances of various human rights violations captured in real world situations and surroundings. This allows for more realistic, and thus more reliable comparisons in different application scenarios. This data is used in various experiments throughout this thesis.

A new benchmark for predicting human rights violations from images. (Chapter 4) Given the nature of the target task, this chapter particularly investigates inner working behind transfer learning for human rights violation recognition. To this end, we examine the transfer learning problem of applying classifiers trained on *everyday objects/scenes* to human rights violations, by comparing their performance to HRVR-trained classifiers (which do not experience any transfer learning) at the same task. We show that there is a notable performance gap between everyday objects/scenes image-trained and HRVR-trained classifiers, and that the visual recognition of human rights violations poses a challenge at a higher level for the well studied representation learning methods.

Analysis of equivalence of various CNN image representations. (Chapter 5) We explore whether two representations, for example two different parametrizations of a CNN, two different CNN architectures, or CNNs trained to classify objects and scenes, share the same visual information or not. This allows us to see how well object-centric and scene-centric CNN features can be combined for solving the task of predicting potential human rights violations. We show that even though object-centric and scene-centric CNNs do not share the same informative regions relevant to their predictions, their feature fusion trail their individual

counterparts in most testing scenarios. Additionally, a practical application of this research, a web demo for predicting human rights violations that may be used directly by human rights advocates and analysts, is developed.

Recognising displaced people from images by exploiting their dominance level. (Chapter 6) This chapter studies the problem of labelling real-world images as either `displaced people` or `non-displaced people`. This is a challenging binary classification problem in the context of human rights investigations, as methods that are based solely on object detection or scene recognition regularly fail to discriminate the encoded visual content of an image that depicts a non-violent situation and the encoded visual content of an image displaying displaced people. We introduce a new method for recognising displaced people by exploiting the overall dominance level of people inside an image.

Harnessing emotional traits for human rights violation recognition in images. (Chapter 7) In this chapter, we demonstrate that classification of certain human rights violations can be improved by integrating the emotional states of persons with a data-driven CNN classifier. To achieve this we propose a novel mechanism capable of characterising an entire image based on all people's emotional states, termed *global emotional traits (GET)*, by utilising two of the continuous dimensions of the Valence, Arousal, and Dominance (VAD) emotional state model [69] that can be associated with human rights content.

Open source development. During the work on this thesis, we have developed and made publicly available multiple open-source projects. Among those are MatDeepRep[1], Human Rights UNderstanding (HRUN) CNNs[2], Human Rights Archive (HRA) CNNs[3], DisplaceNet[4], and GET-AID[5]. Furthermore, several reference implementations [6,7] for Keras framework [8] have been made during the work of this thesis.

[1] https://github.com/GKalliatakis/MatDeepRep
[2] https://github.com/GKalliatakis/Human-Rights-UNderstanding-CNNs
[3] https://github.com/GKalliatakis/Human-Rights-Archive-CNNs
[4] https://github.com/GKalliatakis/DisplaceNet
[5] https://github.com/GKalliatakis/GET-AID
[6] https://github.com/GKalliatakis/Keras-VGG16-places365
[7] https://github.com/GKalliatakis/Keras-Application-Zoo

1.5 Publications

The research conducted in this thesis has resulted in several peer-reviewed publications listed below in chronological order:

- Kalliatakis, G., Ehsan, S., Fasli, M., Leonardis, A., Gall, J. and McDonald-Maier, K. D. (2017). Detection of Human Rights Violations in Images: Can Convolutional Neural Networks help? In *Proceedings of the Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2017*. [41]

- Kalliatakis, G., Stamatiadis, G., Ehsan, S., Leonardis, A., Gall, J., Sticlaru, A. and McDonald-Maier, K.D. Evaluating deep convolutional neural networks for material classification. In *Proceedings of the Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2017*. [45]

- Kalliatakis, G., Ehsan, S. and McDonald-Maier, K.D. A Paradigm Shift: Detecting Human Rights Violations Through Web Images. In *Proceedings of the Human Rights Practice in the Digital Age Workshop, 2017*. [44]

- Kalliatakis, G., Sticlaru, A., Stamatiadis, G., Ehsan, S., Leonardis, A., Gall, J. and McDonald-Maier, K.D. Material Classification in the Wild: Do Synthesized Training Data Generalise Better than Real-World Training Data? Evaluating deep convolutional neural networks for material classification. In *Proceedings of the Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2018*. [46]

- Kalliatakis, G., Ehsan, S., Leonardis, A., Fasli, M. and McDonald-Maier, K.D., 2019. Exploring object-centric and scene-centric CNN features and their complementarity for human rights violations recognition in images. *IEEE Access, 2019*. [43]

- Kalliatakis, G., Ehsan, S., Fasli, M. and McDonald-Maier, K.D. DisplaceNet: Recognising Displaced People from Images by Exploiting Dominance Level. In *Proceedings of the Computer Vision for Global Challenges Workshop, CVPR 2019*. [40]

- Kalliatakis, G., Ehsan, S., Fasli, M. and McDonald-Maier, K.D. GET-AID: Visual recognition of human rights abuses via global emotional traits. *Under review, 2019*. [42]

Chapter 2

Background

While the use of digital images among human rights advocates is becoming more common, innovations are being taken up unevenly, and advocates admit that they tend to utilise opportunistic and adaptive approaches to problem solving instead of purpose-built human rights technology. In this chapter, we review the existing relevant literature that serves as background for the research conducted in this thesis. In order to motivate the uncharted task of visual recognition of human rights violations, we start by examining the role of images in the context of human rights and humanitarian communication. We also explore how and why visual knowledge shapes human rights technology, and a variety of studies related to human rights violations, each of which has benefited from computer vision techniques. Next, we review the literature related to image representation learning which is fundamental in visual recognition, and how it has advanced throughout the years. Finally, we provide a literature review for large-scale image datasets, object detectors, emotion recognition systems, and techniques on transfer learning.

2.1 Introduction

This chapter provides a general introduction to human rights technology as a field, and introduces the previous work that is most pertinent to this thesis. This is followed by an overview of the literature on deep image representations, large-scale image datasets, and transfer learning. Specifically, the structure of the review is the following: We begin, in Section 2.2, with a discussion of the interplay between visuals and human rights advocacy/accountability efforts. Then, in Section 2.3 we briefly present the progression of deep image representations and more specifically Convolutional Neural Networks (CNNs), which are more suitable for dealing with vision problems. We follow up by describing the two most prevalent large-scale image datasets for modern representation learning tasks, ImageNet [12] and Places [119] in Section 2.4, modern object detectors in Section 2.5, and emotion recognition approaches in Section 2.6. Finally, we review literature on transfer learning in Section 2.7. This is of particular relevance to us, as in this thesis we are often concerned with the task of learning in the domain of *everyday objects/scenes*, and transferring this knowledge to the domain of *human rights*.

2.2 Images and Human Rights

Images, moving and still, are undoubtedly powerful, and yet we really do not know exactly how they affect us. The visual shift has been having colossal consequences in the many practices related to the definition and implementation of crucial aspects for human rights. Different individuals–human rights activists, journalists, eye witnesses, practitioners and supporters–as well as various institutions–governments, courts, NGOs, donors and the media–all have been adapting their efforts to consider the visual element in ways that surpass its symbolic objective in human rights practice. Images have been revolutionized from a bare vehicle for advocacy to a vital evidentiary tool and a form of information. Despite this immense interest in the role of visual imagery in human rights advocacy, relatively limited research has documented its influence, while we are left with many questions about how images educate, communicate and relate to the issues of human rights [62].

2.2.1 Human Rights Technology

Over the past decade, the growing use of the term ‘human rights technology’ indicates a field of practice–understood as fact-finding, advocacy, and litigation toward accountability, transparency, and justice–that has gathered attention across multiple disciplines. The origin of this multidisciplinary interest commenced in 2009, at the University of California, Berkeley where an international conference with a diverse mix of academics, practitioners, and technologists

(a) (b) (c)

Figure 2.1: Examples of structure detections in Doro settlement with remote sensing data. Left: ground truth locations of structures; centre: detected structure polygons with adapted network; right: detected structure polygons with basenet. Reprinted from [79].

was held, and debated about the uses of technology for human rights practice as described by Enrique Piracés in [78]. Since then, various foundations, like the MacArthur Foundation, the Ford Foundation, the Oak Foundation, Humanity United, and the Open Society Foundations have adjusted their portfolios to help create the human rights technology field. Currently there are numerous annual workshops on international, regional, and national level which accommodate analysis on the usage of technology for human rights [111]. However, today the vast majority of human rights practitioners utilise technologies which are based on creative or opportunistic variations of general-purpose technologies. There are only a few solutions of purpose-built human rights technology mainly due to the open source nature of the software behind them.

2.2.2 Remote Sensing

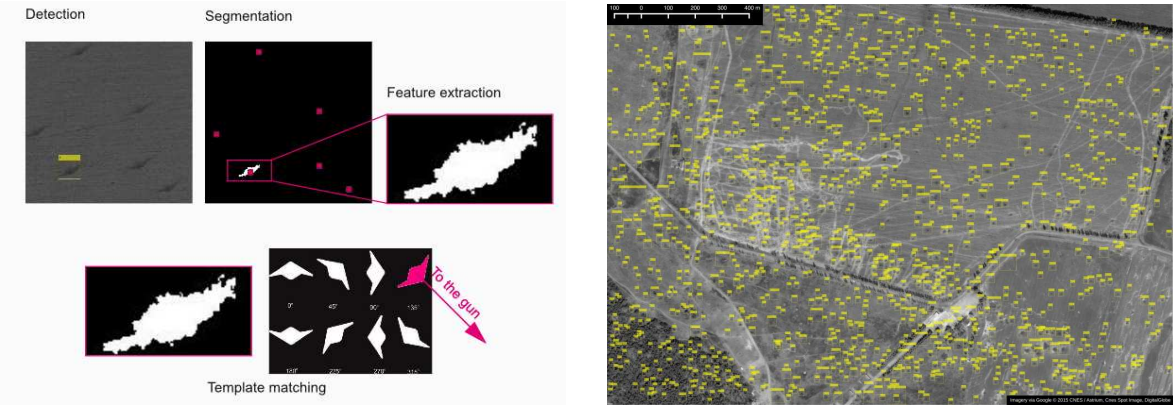
Remote sensing, the science of obtaining information about objects or areas from a distance, typically from aircraft or satellites, is a technology which is increasingly being used to monitor, mitigate and guide humanitarian responses to conflict, human rights violations, and man-made or natural disasters [103, 101]. The use of this technology for studying violent conflict and human rights has increased noticeably over the last decade, and is particularly valuable in difficult-to-reach or dangerous conflict zones where field observations are sparse or non-existent [108]. Recently, Quinn *et al.* presented a case study of experiments using deep learning methods to count the numbers of different types of structures in a refugee or internally displaced people (IDP) settlements in Africa and Middle East [79], which in practice is currently routinely done by human expert analysts. They used annotated, high resolution imagery from thirteen IDP settlements, which were collected by different satellites and/or at different time. Object

detection and region selection was conducted through a Mask-RCNN model [30], by first connecting the input layer to a feature extraction stage of a pre-trained network. Some structure detection examples are shown in Figure 2.1. Their results demonstrate that it is possible to detect a large proportion of structures within settlements. However, the considerable variation in the characteristics of the imagery was also evident in their results. Finally, in order to achieve applicable levels of accuracy when translating generally trained models to specific locations, a semi-automated interactive learning approach has to be followed.

Another system that uses computer vision to analyse images obtained from distance is the *ARtillery Crater Analysis and Detection Engine (ARCADE)* [29]. This prototype scans satellite imagery drawn from Google Maps for signs of artillery bombardment, geocodes artillery blast craters, and calculates the inbound trajectory of projectiles to help automate the process of determining their origin of fire, as illustrated in Figure 2.2a. First, using the Viola-Jones detection algorithm [102]–trained on positive (images containing the craters) and negative (images that do not contain craters) samples–the system flags areas of interest by passing a small-scale window over every part of the image. Then, the system creates a different version of the input images which contain the boxes from the previous step. ARCADE uses active contour segmentation [49] to provide an outline of potential craters. After that, the system uses feature extraction to provide useful data on the latitude and longitude of a crater and estimate its trajectory. Finally, by utilising template matching ARCADE tries to estimate the inbound trajectory of the projectile that created the crater before creating two separate image files which display the areas of the image that appear to correspond with its data on what a crater is, and the image segmentations and outlines of the craters, as seen in Figure 2.2b.

2.2.3 Event Detection in Human Rights Investigations

Several studies are concerned with the detection of particular objects that are of interest to human rights researchers, including tanks, missiles, helicopters, aeroplanes, military vehicles, soldiers, and large crowds [4]. Event detection is another common task within human rights investigations, although it is computationally demanding because of the semantic concept detection involved in a dynamic environment. Another challenge is that the videos relevant to human rights investigations are significantly more complex, while the camera is most of the times unconstrained in time and space. One system that attempts to address these challenges is the *Event Labeling through Analytic Media Processing (E-LAMP)* [98]. An operator provides the system with a set of training videos that depict a specific event alongside a set of videos that depict irrelevant actions. Then, E-LAMP analyses these videos for various different features which can be combined into a computational machine learning (ML) model of the relevant event. Then, the system examines a larger collection of videos for additional potential examples



(a) (b)

Figure 2.2: Automating the detection of artillery craters by using the Artillery Crater Analysis and Detection Engine (ARCADE). (a) Automated process of geolocating and obtain trajectory data from suspected blast craters near Amvrosiivka, Donetsk Oblast, Ukraine. Image date 14 September 2014. Reprinted from Google ©2015 DigitalGlobe. (b) Output by ARCADE on crater field at Savur Mohyla, Donetsk Oblast, Ukraine. Image date 14 September 2014. Imagery via Google © 2015 CNES / Astrium, Cnes Spot Image, DigitalGlobe.

of this particular model, and returns a set of videos that it anticipates match the event in question. Finally, in order to establish a classifier for the particular action, the operator has to confirm whether the suggested matches are correct or not, and the system goes through the collection again after taking into consideration the operator's verdict. The classifier can be used to detect duplicates or near-duplicates, as illustrated in Figure 2.3, but needs to be adjusted before being deployed to different context. Results are returned with probability calculations–the higher the score, the more confident the system is for its prediction.

2.2.4 Event Reconstruction

Reconstructing events in time and space using computer vision, is one more area which computer scientists are increasingly interested in. Event reconstruction can help investigators decipher what happened during an event alongside in which context the event occurred. Initial work has focused on developing tools for displaying several synchronised videos of an event at the same time, allowing event observation from diverse angles. For instance, Forensic Architecture [107] makes use of diverse sources including photos, cell phone audio and video, satellite imagery, digital mapping, and security camera and broadcast television footage, to carefully reconstruct the scene of a violation as a virtual three-dimensional architectural model.

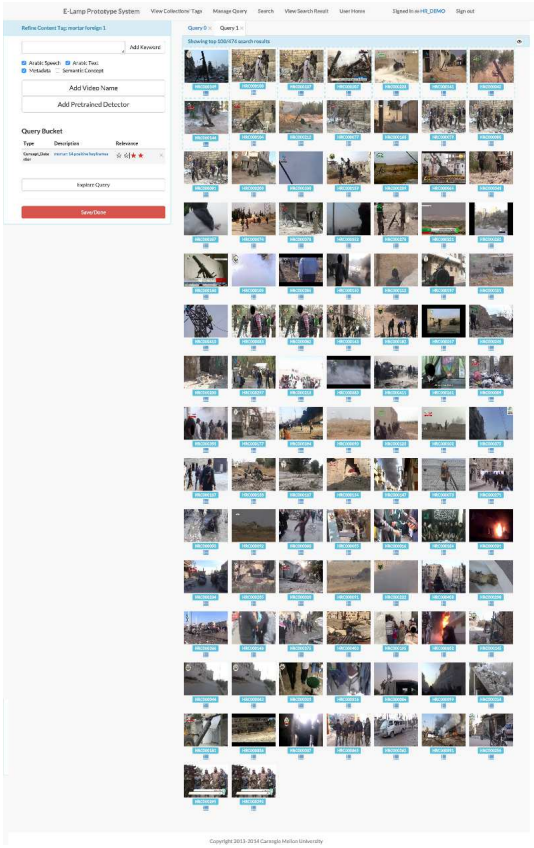


Figure 2.3: Event Labeling through Analytic Media Processing (E-LAMP) search results returned from classifier. Top 100 Mortar Launcher Videos out of 476 total. Reprinted from E-LAMP [4].

2.2.5 Facial Recognition Systems

One more area of computer vision that potentially has a plethora of potential uses in the human rights domain is facial recognition. Over the past decade, researchers have developed sound methods that make face detection relatively standard even though all human faces are unique. A much more complicated task that can lead to issues in lower-resolution images is face recognition; determining if faces from different images belong to the same person. One could imagine human rights investigators using those facial detection and recognition methods to identify perpetrators, victims from recorded violent conduct, or even eyewitnesses who could provide testimony in an investigation [24]. However, in practice there are great ethical concerns, negative applications, and various technical limitations that make its use in the human rights domain less feasible. First, the low resolution nature of most videos retrieved from social media cannot produce enough fine-grained data for facial recognition systems to estimate sufficient facial characteristics in order to provide relevant matches. Second, the majority of the available

facial recognition systems underperform when identifying people of African ancestry compared to European descendants, mainly as a consequence of the training data attributes. Third, for many videos in the human rights context, faces are obscured (head coverings *etc.*) leaving only a small portion of the face visible [37], while damage from blunt force trauma, drowning, burns, and other factors significantly alter the characteristics of the face to the point that visual recognition even by family members becomes difficult [71].

2.3 Deep Image Representations

Image representations have been a primary focus of computer vision research for many years [68]. Most image understanding methods rely on various image representations such as histogram of oriented gradients (SIFT [64] and HOG [11]), sparse [112] and local coding [104], bag of visual words [10], super vector coding [121], textons [57], VLAD [35], Fisher Vectors [77], and, lately, deep neural networks, notably of the convolutional family [54, 115, 86]. For local invariant feature detection and matching, SIFT gained huge popularity before deep image representations due to its strong detector and highly distinctive descriptor. The algorithm is divided into four main stages: scale-space extrema detection, keypoint localisation, orientation assignment and keypoint descriptor computation. The intuition behind SIFT relies on finding 'keypoints' in an image and then computing a 128-dimensional descriptor around that point to summarise local gradient histograms information in a scale and rotational invariant way. HOG has been applied extensively in object recognition areas. HOG captures features by counting the occurrence of gradient orientation. Traditional HOG divides the image into different cells and computes a histogram of gradient orientations over them. However, these points are based on photometric considerations (*e.g.* gradients), and are not *semantically* consistent, they do not consistently detect parts of objects.

An image x is associated with an image representation ϕ that encodes the visual content of x in a way that the predictor ψ performs a prediction of a label $\hat{y}$ which can be discrete/categorical for classification or continuous for regression problems. Typically, both the encoder ϕ and predictor ψ are parametrised by σ_ϕ and σ_ψ respectively, as shown in Figure 2.4. Essentially, the aim of the prediction pipeline is to have an encoder which reflects the prior knowledge of the input domain, either by hand-crafting it, or by reusing an encoder trained on different tasks, also known as *transfer learning*.

It is possible to categorise image representations based on different aspects, but one of the most accepted categorisations is between 'handcrafted', such as SIFT and HOG for which σ_ϕ is constant, and end-to-end trained image representations, where σ_ϕ is trained. These modern image representations are produced by feed forward artificial neural networks (Deep Neural

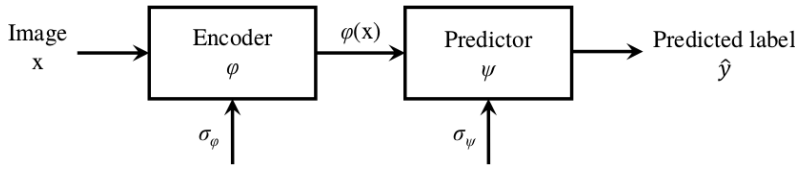


Figure 2.4: A typical image prediction pipeline. An input image x is encoded by the encoder ϕ to obtain a representation $\phi(x)$ which is used by the predictor ψ to predict the label $\hat{y}$. σ_ϕ and σ_ψ parametrise the encoder and predictor respectively in case of a trained feature representation.

Networks DNNs), hence referred to as 'deep representations'. In the context of an image classification task, deep image representations are usually trained using the maximal likelihood method. For a given dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ for set of image x_i and its label y_i, the ML trained parameter vector $\sigma^* = (\sigma_\phi^*, \sigma_\psi^*)$ is estimated as follows:

$$\sigma^* = argmin \frac{1}{N} \sum_{(x_i,y_i)\in D} -\log P(y_i|x_i, \sigma) \tag{2.1}$$

where the probability is estimated as follows:

$$P(y|x, \sigma) \sim P(y|\psi(\phi(x; \sigma_\phi); \sigma_\psi)) \tag{2.2}$$

In the context of this thesis, we have heavily used Convolutional Neural Networks (CNNs), which is why we only focus on literature concerning deep image representations as background.

2.3.1 Convolutional Neural Networks (CNNs)

One of the first neural networks in computer vision is 'Neocognition' which dates back to the 1980s [23]. This hierarchical network consists of many layers, and variable connections between nodes in nearby layers. Initially the lower level nodes extract the local features of the input, and are progressively unified with more general features. At that time, this model was influenced by the groundbreaking studies of the visual cortex of cats by Hubel and Wiesel [33]. The network becomes robust to deformation scales and translations in the position of the inputs, thanks to the pooling layers. After training, the network is ready to perform simple pattern recognition.

Nine years later, influenced by Neocognition, the convolutional model was re-established by LeCun *et al.* [55] and was successfully applied to handwritten digit recognition. This network was made up of two convolutional layers with 12 filters each, and two fully connected layers with 30 hidden units (not present in the Neocognition model). It was trained with the back-propagation algorithm on 7291 training samples of resolution 16×16 pixels having 9760 free parameters overall.

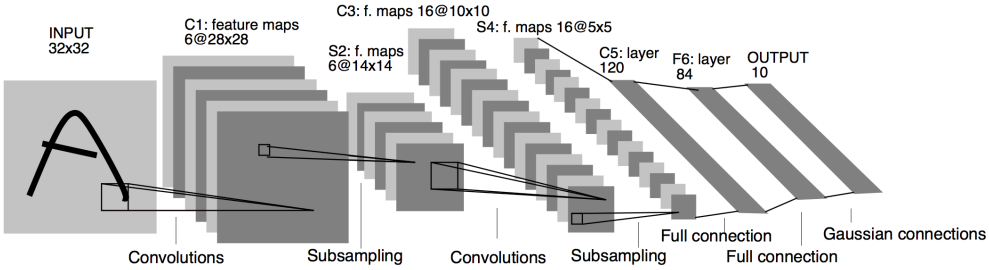


Figure 2.5: Architecture of LeNet deep convolutional neural network, figure reprinted from the original paper [56].

2.3.2 LeNet

Throughout the years, the network was further upgraded to the LeNet model illustrated in Figure 2.5. The main advancement in LeNet is the establishment of *max pooling*–at that time it was parametrised with a limited number of learnable parameters–which advances spatial sub-sampling and allows more filters to be used. Following this approach, more hidden units in the fully connected layer were feasible alongside a classifier on top. This network processed input images of 32×32 pixels with $60 \cdot 10^3$ free parameters and was trained on $60 \cdot 10^3$ training samples. Although LeNet has achieved state-of-the-art results on the Modified National Institute of Standards and Technology (MNIST) dataset, generalising to real-world vision problems was hampered by two severe limitations. First, it required thousands of iterations to converge with stochastic gradient descent (SGD), while the computational power was limited at that time, and second due to millions of parameters of a typical CNN that lead to overfitting of the small-scale MNIST dataset.

2.3.3 AlexNet

The first showcase of the power of CNNs in computer vision happened in 2012 when Krizhevsky *et al.* [54] trained AlexNet for the ImageNet challenge [84]. Their network architecture–reproduced in Figure 2.6–consists of five convolutional layers, followed by three fully connected layers[1]. The non-linear operation used between layers is the rectifier $f(x) = max(0,x)$, frequently referred to as *rectified linear unit (ReLU)*, while max-poolings are used to downsample the feature maps and gradually add negligible invariance. In order to avoid overfitting, data augmentations are applied during training (colour jitterings, rotations, random croppings), as well as dropout–the activation of each neuron is zeroed stochastically with 50% probability [93]. Dropout is used for the first two fully connected layers and must be noted that without it,

[1] A fully connected layer is simply a convolutional layer where each filter is the same size as the input

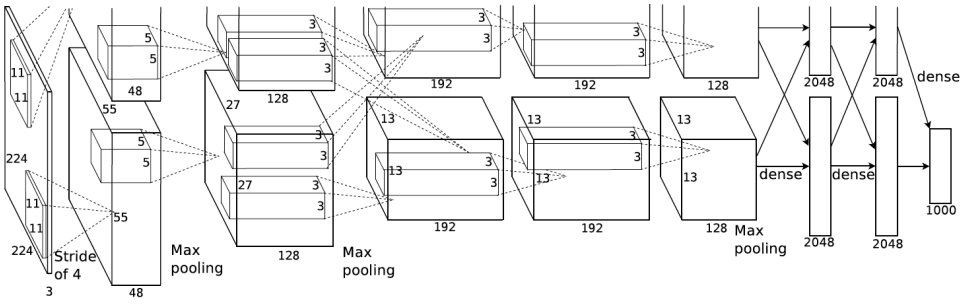


Figure 2.6: Architecture of AlexNet deep convolutional neural network reproduced from [54] for classification of the ILSVRC 2012 dataset.

the network exhibits significant over-fitting. The output of the network is a 1000-D softmax vector corresponding to the probability of an image belonging to each class. The network was trained with the trainval set of ILSVRC-2012 [84] which consists of over 1 million single labelled images corresponding to one of 1,000 object classes. The network is trained on two graphic processing unit (GPU) cards by putting batches of images through the network, while their soft-max vectors are used in conjunction with the correct labels in a logistic loss. The filters are being updated by back-propagating the gradient of this logistic loss. AlexNet contains $61 \cdot 10^6$ free parameters and training with randomly initialised weights took around a single week. This network achieved a top-5 error rate of 15.3% on the test set of ILSVRC-2012, compared to 26.2% using a shallow Fisher Vector representation. Compared to other deep models, this network significantly increases the number of free parameters while it is trained in a fully supervised manner. It is believed that the main reason behind its large performance boost compared to previous approaches is the increased size of the training set and application of new regularisation methods.

2.3.4 VGGNet & GoogLeNet

Since AlexNet, more powerful network architectures have emerged to further advance image classification with CNNs. In 2014, VGGNets [90] and GoogLeNet [95] achieved considerable performance improvement by creating deeper representations–CNNs with more layers. Specifically, Simonyan and Zisserman [90] achieved state-of-the-art performance at the ILSVRC localisation challenge by introducing VGGNet. This network replaces large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. Reducing volume size is handled by max pooling, while all hidden layers are equipped with the rectification (ReLU) non-linearity. Three fully connected (FC) layers follow a stack of convolutional layers: the first two have 4096 channels each, the

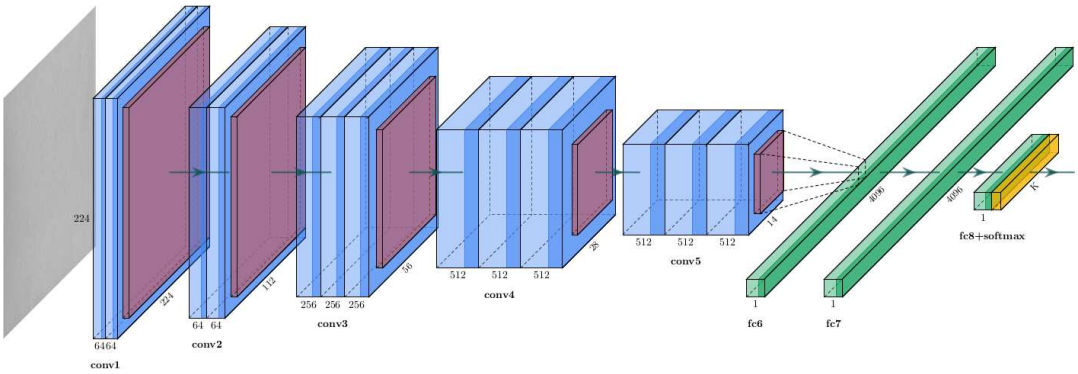


Figure 2.7: Architecture of VGG16 network [90]. In this network all filter kernels have size of 3×3 and max pooling layers are placed after each 2 convolutions.

third performs 1000-way ILSVRC classification and thus contains 1000 channels, as illustrated in Figure 2.7.

In Google Inception network (frequently referred to as GoogLeNet) Szegedy *et al.* [95] propose to use *inception blocks*, the core concept of a sparsely connected architecture, where several kernels with multiple receptive field sizes for convolution (5×5, 3×3, and 1×1) are concatenated into a single output vector forming the input of the next stage, similar to Network in Network (NIN) [59]. In order to overcome the gradient vanishing issue–as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small–auxiliary supervisions are applied on several intermediate layers. They essentially applied softmax to the outputs of two of the inception modules, and computed an auxiliary loss over the same labels. The total loss function is a weighted sum of the auxiliary loss and the real loss. Also, in comparison to the AlexNet architecture, GoogLeNet has only one fully connected layer (the classifier) after average pooling, to go from a $7 \times 7 \times 1024$ volume to a $1 \times 1 \times 1024$ volume, as seen in Figure 2.8. Although GoogLeNet uses 9 inception modules in the whole architecture, and over 100 layers in total, it drastically reduces the number of free parameters by 12 times compared to AlexNet, while it achieves 6.7% top-5 error rate on ILSVRC-2014.

2.3.5 ResNet

Following the establishment of VGGNet and GoogLeNet, researchers soon discovered that increasing network depth by simply stacking layers together saturates the performance due to vanishing gradient problem, despite a clear interrelation between network depth and image classification performance. A network that solves the vanishing gradient problem is the Residual Network (ResNet) [31] architecture which held state-of-the-art results on ILSVRC in 2015. This 'ultra-deep' architecture significantly increases the number of convolutional layers;

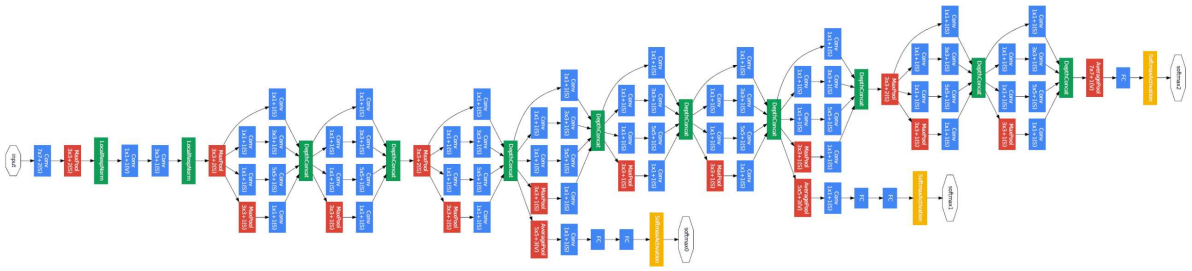


Figure 2.8: Architecture of GoogLeNet reproduced from the original paper [95]. The network uses combinations of inception modules, each including some pooling, convolutions at different scales and concatenation operations.

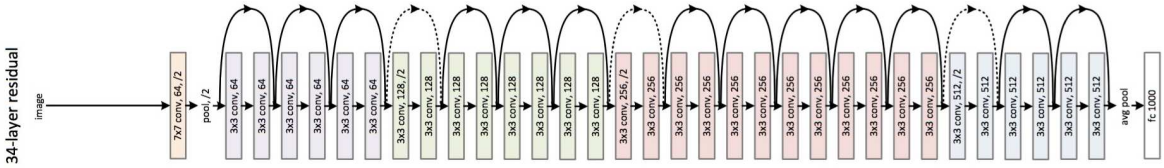


Figure 2.9: Architecture of ResNet reprinted from the original paper [31]. Residual units are extensively used in this network, while at the end average pooling is used to aggregate global feature representations.

training this deep network is possible thanks to skip connections and heavy batch normalisation. Each 'residual unit' can be expressed in a general form:

$$\begin{aligned} y_n &= h(x_n) + F(x_n, W_n) \\ x_{n+1} &= f(y_n) \end{aligned} \tag{2.3}$$

where x_n and x_{n+1} are the input and output of the nth unit respectively, and F is a residual function. The model is trying to learn this residual function which keeps most of the information and produces only slight changes. Consequently, patterns from the input image can be learned in deeper layers, while no additional parameters are added and the computational complexity does not increase. ResNet consists of 152 convolutional layers with 3x3 filters using residual learning by block of two layers, as shown in Figure 2.9. He *et al.* showed that ResNet with 50/101/152 layers were more accurate than a 34 layers plain network.

2.3.6 DenseNet

Another network that was proposed to solve the vanishing gradient problem is DenseNet [32]. DenseNet was developed by Gao *et al.* in 2017, and consists of densely connected convolutional layers. The output of each layer is connected with all subsequent layers in a dense block. DenseNet consists of several dense blocks and transition blocks, which are placed

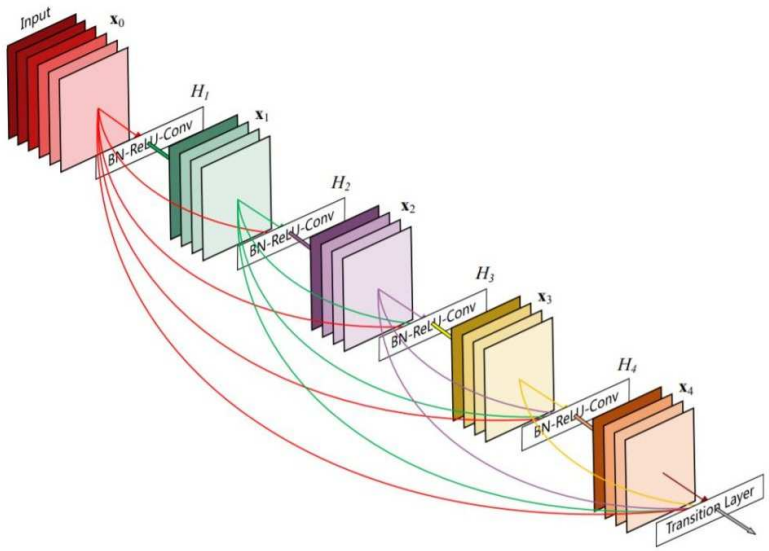


Figure 2.10: Architecture of DenseNet reproduced from the original paper [32]. A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input.

between two adjacent dense blocks. The proposed architecture is shown in Figure 2.10. While ResNet explicitly preserves information through additive identity transformations, DenseNet connects each layer to every other layer in a feed-forward fashion, thus feature maps of all preceding layers are used as inputs into all subsequent layers. As DenseNet concatenates previous layers features instead of adding them, it manages to explicitly differentiate between information that is added to the network and information that is preserved.

2.3.7 Interpreting Deep Visual Representations

As deep neural networks surpass humans on various visual tasks, the need for deeper understanding of the underlying mechanisms of these representations became apparent. Deep neural networks are often criticised as being black boxes that lack interpretability due to their large amount of model parameters. This absence of interpretability can considerably limit the usage of complex models for wider computer vision applications. Recently, there has been a growing number of works on understanding deep visual representations. The related work can be categorised as following two main aspects: *visualising deep representations* and *analysing the properties of deep representations*.

The behaviour of CNNs can be visualised either by sampling image patches that maximize activation of hidden units [27, 115, 117] or by using modified backpropagation to generate or identify salient image features [7, 66, 88]. Mahendran and Vedaldi have shown that backpropagation along with a natural image prior can be used to invert a CNN activation [66], while

an image generation network can be trained to invert deep features by synthesising the input images [14]. These particular visualisations expose the learned image patterns in a deep visual representation, as well as provide a qualitative sign to the interpretability of units. In [117], Zhou *et al.* introduced a quantitative measure of interpretability to determine which individual units behave as object detectors within a network trained to classify scenes.

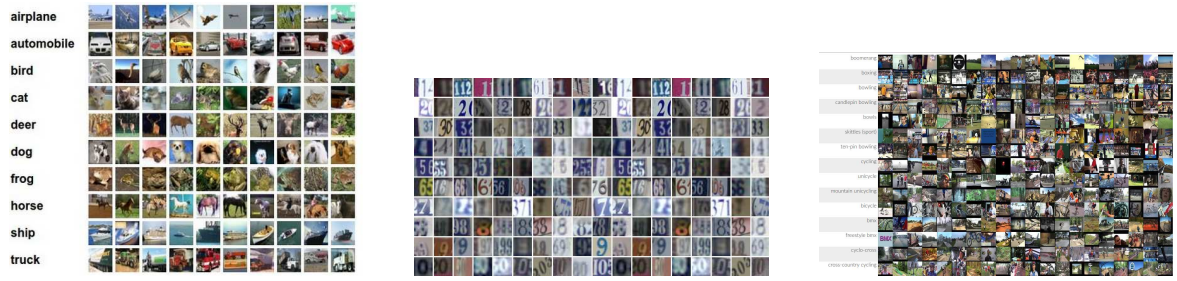
Much research has also focused on studying the power of CNN layer activations to be utilised as generic visual features for classification tasks [2, 87]. Other authors have noted interesting properties of deep representations. Another notable work, by Szegedy *et al.* [96] discovered that it is easy to fool deep neural networks by making small adjustments to the input image. Another interesting observation made in [58] is that many units converge to the same set of representations after training. Finally, the question of how visual representations generalise has been explored by showcasing that a CNN can successfully fit a random labelling of training data even under explicit regularisation [116].

2.4 Notable Datasets

As digitisation of society increases, more and more of people's activities are being recorded. While our computers are increasingly networked together, it has become easier to centralize these records and curate them into a dataset appropriate for deep learning applications, as stated by Goodfellow *et al.* in [28]. In the first decade of the 2000s, sophisticated datasets containing tens of thousands of examples, such as the CIFAR-10 dataset [53] (shown in Figure 2.11a) started to be constructed. Towards the end of that decade and throughout the first half of the 2010s, significantly larger datasets, containing hundreds of thousands to tens of millions of examples, completely changed what was possible with deep learning. These datasets included the public Street View House Numbers dataset [72], and the Sports-1M dataset [48], shown in Figure 2.11b and Figure 2.11c respectively.

2.4.1 ImageNet

The ImageNet database [12], is the result of a collaboration between Stanford University and Princeton University and has become the standard benchmark for large-scale object recognition since its first appearance in 2009. ImageNet consists of over 15 million labelled high-resolution images, belonging to roughly 22,000 categories, which are collected from the web and labelled by human workers using Amazon Mechanical Turk (MTurk), a crowdsourcing marketplace that makes it easier for individuals and businesses to outsource their processes and jobs to a distributed workforce who can perform these tasks virtually. One of the most common



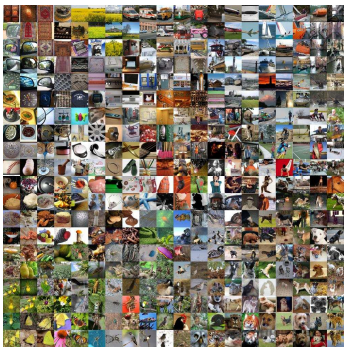
(a) CIFAR-10 **(b)** SVHN **(c)** Sports-1M

Figure 2.11: Example inputs from different modern datasets. (a) The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. (b) Street View House Numbers (SVHN) dataset consists of over 600,000 digit images in 10 classes. (c) The Sports-1M dataset contains 1,133,158 video URLs which have been annotated automatically with 487 Sports labels using the YouTube Topics API.

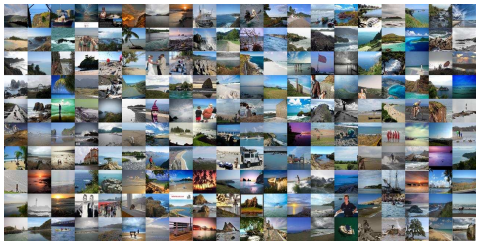
tasks on MTurk is labelling photos that are used to train deep learning models. ImageNet is the backbone of the annual competition called *ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)* which was established in 2010. ILSVRC uses a subset of ImageNet with approximately 1000 images in each of 1000 categories. Overall, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. Examples of validation images can be seen in Figure 2.12a. ILSVRC follows in the footsteps of the PASCAL VOC challenge [19], established in 2005, which set the criterion for standardised evaluation of recognition algorithms in the form of yearly competitions. Before long, it became increasingly common within the computer vision community to treat image classification on ImageNet as an intermediate procedure for training deep CNNs to learn proper general-purpose features. This practice of first training a CNN on ImageNet (*i.e.*, pre-training) and then adapting these generic features for a new target task (*i.e.*, fine-tuning) has become the *de facto* standard for solving a wide range of computer vision problems such as image classification, object detection, action recognition, image segmentation, image captioning, human pose estimation, and others.

2.4.2 Places

Similar to ImageNet object-centric dataset, Places database [119] is a scene-centric repository of scene photographs, labelled with scene semantic categories, comprising about 98% of the type of places a human can encounter in the world. Places consists of 10 million scene photographs, labelled with 434 scene semantic categories, making it the other large-scale image dataset where researchers are able to train CNNs from scratch besides ImageNet. Image samples from the coast category of Places are shown in Figure 2.12b. Places follows in the footsteps of the SUN database [110], which has a rich scene taxonomy, by inheriting the



(a) samples from the ImageNet validation set



(b) 'coast' category of the Places database

Figure 2.12: Image samples from large-scale image databases. (a) Samples of different categories from the ImageNet validation set, figure has been reproduced from [47]. (b) Samples from the 'coast' category of the Places database.

same list of scene categories. As part of the ILSVRC challenge, Zhou *et al.* released the *Places365-Challenge* subset for the Places Challenge, which was held in conjunction with the European Conference on Computer Vision (ECCV) in 2016, with a total of 8 million training images, 50 images per class for validation and 900 images per class for testing. Since their introduction, Places-CNNs have been deployed as an intermediate procedure for training deep CNNs to learn generic visual features for scene-based visual recognition tasks, as a direct replacement to object-centric ImageNet pre-trained CNNs.

2.5 Object Detection

One of the most improved areas of computer vision in the past few years is object detection, the process of determining the instance of the class to which an object belongs and estimating the location of the object. This section is an overview of current object detection methods. This is of particular relevance to us, as in the last two chapters of this thesis (Chapter 6 and Chapter 7) we are exploiting object detection as a core component of our proposed methods. Object detectors can be split into two main categories: *one-stage detectors* and *two-stage detectors*.

2.5.1 One-Stage Detectors

The OverFeat model [86] which applies a sliding window approach based on multi-scaling for jointly performing classification, detection and localization was one of the first modern one-stage object detectors based on deep networks. More recently YOLO[63, 22] and SSD [80, 81] have revived interest in one-stage methods, mainly because of their real time capabilities,

although their accuracy trails that of two-stage methods. One of the main reasons being due to the class imbalance problem [60].

2.5.2 Two-Stage Detectors

The leading model in modern object detection is based on a two-stage approach which was established in [99]. The first stage generates a sparse set of candidate proposals that should contain all objects, and the second stage classifies the proposals into foreground classes or background. Region with CNN Features (R-CNN), a notably successful family of methods [25, 26] enhanced the second-stage classifier to a convolutional network, resulting in large accuracy improvements. The speed of R-CNN has improved over the years by integrating region proposal networks (RPN) with the second-stage classifier into a single convolution network, known as the Faster R-CNN framework [82].

In the context of this thesis, we adopt the one-stage approach of RetinaNet framework [60] which handles class imbalance by reshaping the standard cross entropy loss to focus training on a sparse set of hard examples and down-weights the loss assigned to well-classified examples.

2.6 Emotion Recognition

Recognising people's emotional states from images is an active research area among the computer vision community. This is of particular relevance to us, as in the last two chapters of this thesis (Chapter 6 and Chapter 7) we are exploiting a specific emotion representation approach as a core component of our proposed methods. One established categorisation of emotion recognition methods in the literature is to split them into two categories based on how emotions are represented: *discrete categories* and *continuous dimensions*.

2.6.1 Discrete Categories

Most of the research in computer vision to recognise people's emotional states is explored by facial expression analysis [20, 17], where a large variety of methods have been developed to recognise the 6 basic emotions defined in [16]. Many of these methods are based on a set of specific localised movements of the face, called *Action Units*, in order to encode the facial expressions [21, 9]. More recently emotion recognition systems based on facial expressions use CNNs to recognise the Action Units [20].

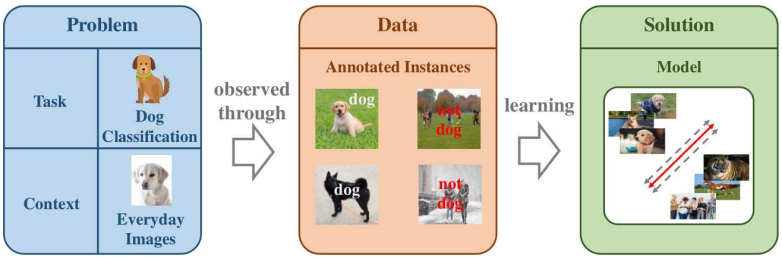


Figure 2.13: Problem solving with deep learning. Problem consists of a task and a context, and can be observed through annotated data. A solution is learnt from those data throughout a learning process.

2.6.2 Continuous Dimensions

Instead of recognising discrete emotion categories, this family of methods use the continuous dimensions of the VAD *Emotional State Model* [69, 70] to represent emotions. The VAD model uses a 3-dimensional approach to describe and measure the emotional experience of humans: Valence (V) describes affective states from highly negative (unpleasant) to highly positive (pleasant); Arousal (A) measures the intensity of affective states ranging from calm to excited or alert; and Dominance (D) represents the feeling of being controlled or influenced by external stimuli. In recent times the VAD model has been utilised for facial expression recognition [92].

In the context of this thesis, we adopt the tridimensional model of affective experience alongside a joint analysis of the person and the entire scene in order to recognise rich information about emotional states, similar to [52].

2.7 Transfer Learning in Computer Vision

In deep learning context, *problems* are abstract concepts observed through the *data* which consists of *instances* and associated *labels* to learn from, while the *solutions* are considered to be the parameters of the model that will be learned for solving the problem. An example can be seen in Figure 2.13 for the dog classification task in the context of everyday images. The data consists of image samples together with their corresponding ground truth labels relevant to the existence or absence of dogs in images. Image instances can be named as positive or negative and are mainly utilised in the form of extracted features. The solution refers to a computational model capable of discriminating the positive instances from the negative ones by assigning a label to test instances.

Transfer learning and domain adaptation refer to the situation where a model is learnt in one setting (*i.e.*, distribution P_1), and is exploited to improve generalization in another setting (say distribution P_2). The transfer process begins with a (a) target task to be learnt in a target

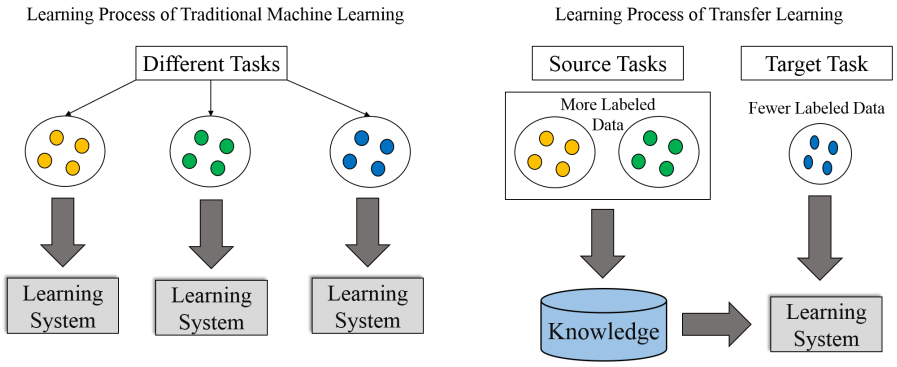


Figure 2.14: The key elements of traditional machine learning and transfer learning, reproduced from [76].

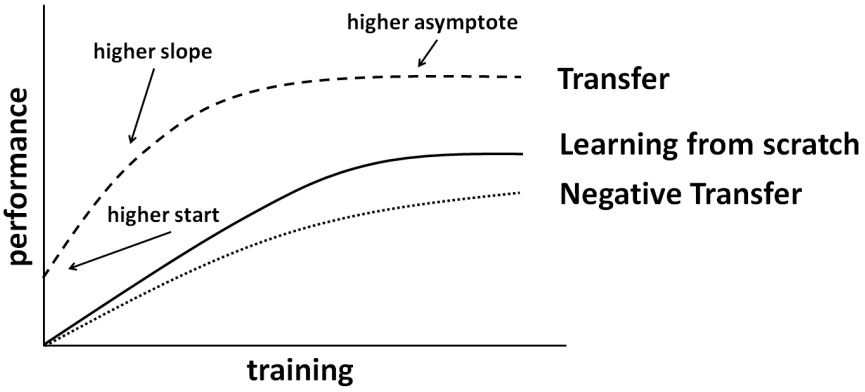


Figure 2.15: The benefits of transfer learning reproduced from [73]. The three types of performance improvement aspirations from transfer learning. The x-axis is the number of training instances for the target problem.

context; (b) a set of solutions to the source tasks (already learnt in the source contexts); (c) the transfer of knowledge based on the similarity between the target and source tasks. Figure 2.14 illustrates the difference between the traditional machine learning process and the transfer learning technique. This is commonly understood in a supervised learning context, where the input is the same but the target may be of a different nature. If there is significantly more data in the first setting (sampled from P_1), then that may help to learn representations that are useful to quickly generalize from only very few examples drawn from P_2. This happens because many visual categories share low-level notions of edges and visual shapes, changes in lighting, *etc*. There are extensive literature reviews on the topic [76, 106].

Recent works have focused on incorporating transfer learning into deep visual representations, to combat the problem of insufficient training data. Pre-training CNNs on ImageNet or Places has been the standard practice for other vision problems. However, features learnt in pre-trained models are not perfectly fitted for the target learning task. Using the pre-trained

network as a *feature extractor* [87, 13, 119] or *fine-tuning* the network [26, 74] have become a frequently used method to learn task-specific features, while extensive efforts have been made to perceive transfer learning itself [5, 34, 94, 114].

It is possible to define three measures by which transfer might improve the effectiveness of learning as discussed in [73, 97]. We list them below referring to Figure 2.15

Higher start. Knowledge transfer approach performs much better compared to learning from scratch, even with very few target instances.

Higher slope. The performance of transfer learning grows faster when additional target instances are introduced to the learning process.

Higher asymptote. Final performance of the transfer method is preferable to the learning target problem alone.

2.8 Summary

In this chapter, after introducing advancements of vision-based technology in the field of human rights, we have provided an overview of computer vision systems which are being used today by human rights practitioners and investigators. Then, the state-of-the-art deep image representations that are prevalent in most modern computer vision systems are thoroughly discussed. After that, we have discussed the two main large-scale image datasets that will be used throughout this thesis namely the ImageNet and Places, before looking at relevant work on object detection and emotion recognition. Finally, we have discussed how we can transfer knowledge learnt in one context to another context for problem solving with image representations. In the next chapter, Chapter 3, we describe the image datasets that we have developed for recognising human rights violations 'in the wild'.

Chapter 3

Datasets for Human Rights Violation Recognition

Large, labelled image datasets are the driving force for novel visual recognition models and progress made for various visual recognition tasks. But what is necessary to achieve expert-level recognition with a deep learning algorithm? In the case of supervised learning, the problem is two-fold. First, the algorithm must be suitable for the task at hand–such as CNNs for large-scale visual recognition, described in Section 2.3.1. Second, the algorithm must have access to a training dataset of appropriate coverage and density. In all cases, a dataset is a collection of examples, which are in turn collections of features. Whereas most image datasets have focused on object or scene categories, a human rights-specific image dataset does not currently exist. This limits the application of powerful deep learning technology on specific domains like the human rights advocacy field. We strive to reconcile this gap by developing the first ever image datasets for human rights violation recognition. We also discuss the challenges encountered during this process compared to standard image collection procedures presented in the literature. We believe our image datasets will facilitate future research on practical visual recognition tasks related to human rights, fine-grained visual classification, and imbalanced learning fields.

3.1 Introduction

In this chapter we present the construction of the image datasets that will be used in the subsequent chapters of this thesis, and the human rights violation recognition benchmarks in detail. We construct three novel image datasets for human rights-specific purposes: (i) `Human Rights UNderstanding` (Section 3.2); (ii) `Human Rights Archive` (Section 3.3); (iii) `HRA-Binary` (Section 3.3.5). Though they are all datasets containing human-rights-violation-related images, they were obtained from different sources and thus contain variations in the acquisition procedure which we will look into.

3.2 The Human Rights UNderstanding (HRUN) Dataset

We introduce the `Human Rights UNderstanding Dataset` which is the first-ever dataset of images and annotations regarding human rights violations. Our objective is to construct a well-sampled image database in the domain of human rights violations, which will be used to assess classification performance of CNNs.

First, the keywords, with a view to formulate the query terms, were collected in collaboration with specialists in the human rights domain, such as United Nations High Commissioner for Refugees (UNHCR) and Office of the High Commissioner for Human Rights (OHCHR). This happens in order to include multiple query terms for every 'targeted class'. For instance, for the class `police violence` the queries 'police violence', 'police brutality' and 'police abuse of force' were all used for retrieving image samples. Work commenced with the Flickr photo-sharing website, but in a short time, it became apparent that its limitations resulted in a huge number of irrelevant results returned for the given queries, as discussed in [44]. This happens because Flickr users are authorised to tag their uploaded images without restriction. Subsequently, there are situations where the given keyword was 'armed conflict' and the majority of the returned images showcased military parades. Another similar example was with the keyword 'genocide' where the returned results included protesting campaigns against genocide, something that may be consider close to the keyword, but it can not serve our purpose. Another shortcoming was the case when people massively tagged an image deliberately incorrectly in order to acquire an increased number of hits on the photo-sharing website.

Consequently, Google and Bing search engines were chosen as better alternatives. Images were downloaded for each class using a python interface to the Google and Bing application programming interfaces (APIs), with the maximum number of images permitted by their respective API for each query term. All exact duplicate images were eliminated from the

(a) Child Labour

(b) Police Violence

(c) Child Soldiers

(d) Refugees

Figure 3.1: Example class images retrieved from Google and Bing search engines for different human rights violation keywords. From top to bottom row: *child labour*, *police violence*, *child soldiers*, *refugees*. Notice that finding relevant instances is challenging: returned samples principally are related to the corresponding human right violation, however they do not depict the particular action or situation that causes the violation.

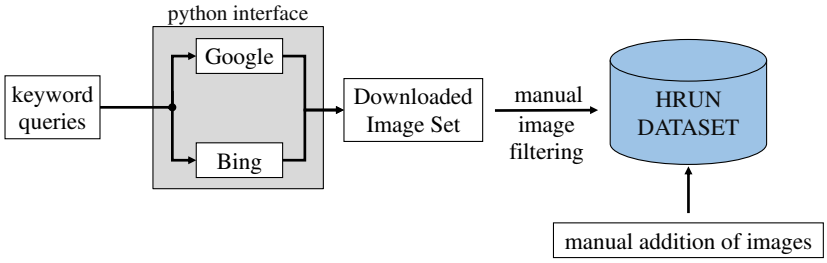


Figure 3.2: Human Rights UNderstanding (HRUN) pipeline overview. We construct a new dataset by combining image samples from Google and Bing search engines with manually added images.

downloaded image set, alongside images regarded as inappropriate during the filtering step as illustrated by Figure 3.1. Nonetheless, the number of filtered images generated was still

query	retrieved		relevant		ratio	
	Google	Bing	Google	Bing	Google	Bing
child labour	99	137	18	5	18%	3.64%
child soldiers	176	159	31	13	17.61%	8.17%
police violence	149	232	10	16	6.71%	6.89%
refugees	111	140	10	39	9.00%	27.85%
aeroplane	170	137	150	135	88.23%	98.54%
car	145	128	123	124	84.82%	96.87%
dog	105	132	101	129	96.19%	97.72%

Table 3.1: The statistics for the image collection procedure of the Human Rights UNderstanding (HRUN) Dataset from search engines. First column corresponds to the query term used for the search. The following four columns correspond to the number of retrieved images which contain that particular class and analysed as relevant respectively, while the last column corresponds to the quantitative relation between those two set of images.

query	retrieved		relevant			HRUN
	Google	Bing	Google	Bing	manually	
child labour	99	137	18	5	77	100
child soldiers	176	159	31	13	56	100
police violence	149	232	10	16	74	100
refugees	111	140	10	39	51	100
Total	535	668	69	73	258	400

Table 3.2: The statistics for the HRUN dataset. Each number corresponds to how many images contain that particular class were retrieved, analysed as relevant, and manually added.

insufficient as shown in Table 3.1. For this reason, there were manually added other suitable images in order to reach the final structure of the HRUN dataset. We finally ended up with a total of four different categories, each one containing 100 distinct images of human rights violations captured in real world situations and surroundings. These are split at random into training, validation and test sets. The entire pipeline used for constructing HRUN dataset is depicted in Figure 3.2. The statistics for this dataset are given in Table 3.2, and example class images are shown in Figure 3.3. Raw data for the images in this dataset are provided at the public repository [38].

3.3 The Human Rights Archive (HRA) Dataset

Although the development of HRUN dataset signals the first ever attempt to produce a high quality image dataset in the context of human rights violations, it quickly become apparent that

(a) Child Labour

(b) Child Soldiers

(c) Police Violence

(d) Refugees

Figure 3.3: Example class images from the `HRUN Dataset`. From top to bottom row: *child labour*, *child soldiers*, *police violence*, *refugees*.

the origin and the verification of those images are considered to be of the utmost importance for human rights practitioners and advocates. For this reason, we revisit the development process of image datasets in the context of human rights violations, by addressing the main drawbacks of our first attempt. We introduce the `Human Rights Archive Database`, a verified-by-experts repository of approximately 3K human rights violation photographs, labelled with human rights semantic categories, comprising a list of the types of human rights abuses encountered at present.

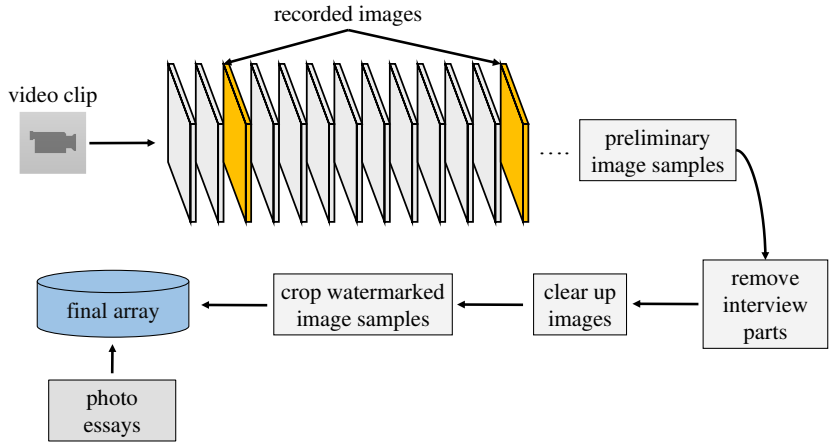


Figure 3.4: Human Rights Archive (HRA) pipeline overview. We construct a new dataset by combining image samples from video clips and image essays, before filtering out all images that do not correspond to the definition of the human right violation category.

3.3.1 Challenges

Human rights violation recognition is closely related to, but radically different from the tasks of object and scene recognition. As an example, one would easily correlate child labour with the task of recognising manual-labour-related tools (*e.g.* hoe and hammer). However, this would clearly be problematic for frequent cases such as adults working with those tools. The same applies for correlating a human right violation with the task of visual place recognition. For this reason, following a conventional image collection procedure is not appropriate for collecting images with respect to human rights violations.

The first issue encountered is that the query terms for describing different categories of human rights violations must be provided by experts in the field of human rights and not by quasi-exhaustively searching a dictionary. The next obstacle concerns online search engines such as Google, Bing or even dedicated photo-sharing websites like Flickr, which returned a huge number of irrelevant results for the given queries of human rights violations as discussed in our study [44], and shown in Table 3.1. The final and most important matter of contention is the ground truth label verification of the images, which commonly is accomplished by crowd-sourcing the task to MTurk. However, in the case of human rights violations, human classification performance cannot be measured by utilising MTurk for the reason that workers are not qualified for such specialised tasks.

3.3.2 Building the Human Rights Archive Dataset

A key question with respect to visual recognition of human rights violations from real-world images arises: *how can this structured visual knowledge be gathered?* The crucial aspects of such unique image database are the origin and the verification of image samples. For this reason, and in order to obtain an adequate number of verified real-world images depicting human rights violations, we turn to non-governmental organizations (NGOs) and their public repositories. The first NGO considered is Human Rights Watch [83] which offers an online media platform[1] capable of exposing human rights and international humanitarian law violations in the form of various media types such as videos, photo essays, satellite imagery and audio clips. Their online repository contains 9 main topics in the context of human rights violations (arms, business, children's rights, disabilities, health and human rights, international justice, LGBT, refugee rights, and women rights) and 49 subcategories. In total, we download 99 available video clips from their online platform. After that, preliminary image samples are being recorded for every video clip with a ratio of 10–one image out of ten frames is recorded. This is done in order to obtain images distinctive enough on a frame to frame basis. Next, all the images that do not correspond to the definition of the human right violation category (mostly the interview parts of the clips) are manually removed. Images with low quality (very blurry or noisy, black-and-white), clearly manipulated (added text or borders, or computer-generated elements) or otherwise unusual (aerial views) are also removed. One considerable drawback in the course of that process is the presence of a watermark in most of the video files available from that platform. As a result, all the recorded images that originally contained the watermark had to be cropped in a suitable way. Only colour images of 600 x 900 pixels or larger were retrieved after the cropping stage. In addition to those images, all photo essays available for each topic and its subcategories are added, resulting in 342 more images to the final array. The entire pipeline for collecting and filtering out the images from Human Rights Watch is depicted in Figure 3.4.

The second NGO investigated is the United Nations which presents an online collection[2] of images in the context of human rights. Their website is equipped with a search mechanism capable of returning relevant images for simple and complex query terms. In order to define a list of query terms, we utilise all main topics and their respective subcategories from Human Rights Watch and combine them with likely synonyms. For example, in order to acquire images depicting the employment of children in any work that deprives children of their childhood and interferes with their ability to attend regular school, 'child labour', 'child work' and 'child employment' were provided as queries to the database. In total, we download 8550 preliminary

[1] http://media.hrw.org/
[2] http://www.unmultimedia.org/photo/

(a) arms

(b) child labour

(c) child marriage

(d) detention centres

(e) disability rights

Figure 3.5: Example class images from the `HRA Dataset`. From top to bottom row: *arms*, *child labour*, *child marriage*, *detention centres*, *disability rights*.

image samples by utilising the list of query terms. We follow the same approach as Human Rights Watch in order to filter out the images. First, we manually remove all the images that do not correspond to the definition of the human right violation category. In the case of the United Nations online repository, the majority of the returned images showcased people sharing their testimony at various presentations or panel discussions. We also remove images that are

(a) displaced populations

(b) environment

(c) no violation

(d) out of school

Figure 3.6: Further example class images from the `HRA Dataset`. From top to bottom row: *displaced populations*, *environment*, *no violation*, *out of school*.

black-and-white or otherwise unusual (aerial views). Finally, we add applicable high-resolution images to the database.

3.3.3 Data Analysis

The final dataset contains a set of 8 human rights violations categories and 2847 images, that cover a wide range of real-world situations. 367 ready-made images are downloaded from the two online repositories representing 12.88% of the entire dataset, while the remainder (2480) images are recorded from videos coming out of Human Rights Watch media platform.

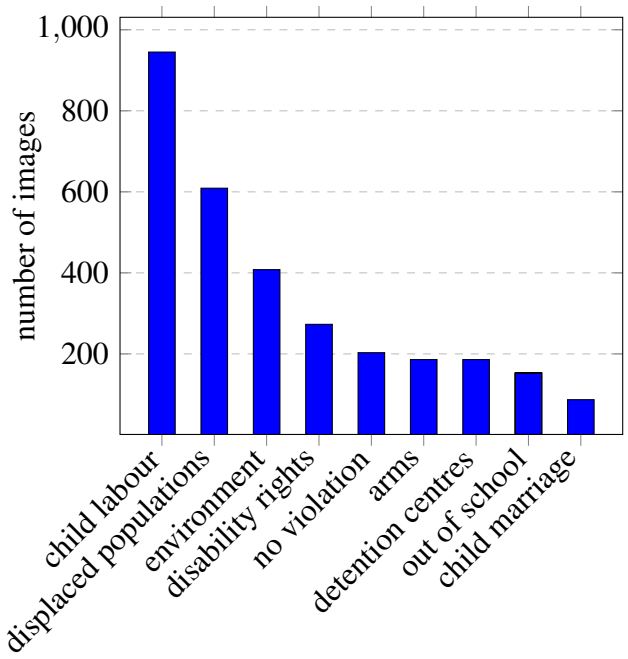
(a) mining **(b)** tobacco workers **(c)** garment workers

(d) Hungary **(e)** Greece **(f)** Central Africa

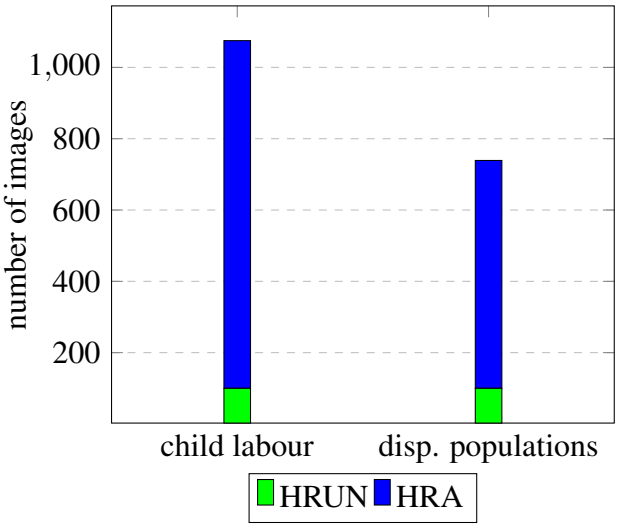
(g) mercury **(h)** tanneries **(i)** contaminated water

Figure 3.7: Image samples from the human rights violation categories of HRA grouped by different situations to illustrate the diversity of the dataset. For each violation category we show 3 labelled images. Top row: `child labour`. Middle row: `displaced populations`. Bottom row: `environment`.

The categories are listed and defined in Table 3.3. Furthermore, 203 instances which are not considered as human rights violations, such as children playing and adult workers mining, have been incorporated into the database in order to assess the classification performance more precisely. Example images in this dataset are given in Figure 3.5 and Figure 3.6. Our human rights-centric dataset differs from our first attempt of Human Rights UNderstanding (HRUN) dataset presented in Section 3.2. That dataset was created by collecting images available on the Internet using online search engines for different manually crafted terms, but the HRA database was created by collecting human rights violations categories from verified sources. Note that, in order to increase the diversity of visual appearances in the HRA dataset, images from different situations or places are gathered, as illustrated in Figure 3.7. Because some violations are reported and documented more than others, the distribution of images is not



(a) Sorted distribution of image number per category in the HRA Dataset. HRA contains 3,320 images from 9 categories.



(b) Comparison of the number of images per violation category for the common two violation categories in HRUN and HRA datasets.

Figure 3.8: Sorted distribution of image number per category in the HRA Dataset and comparison between the common two violation categories in HRUN and HRA datasets.

uniform between the classes of the database, as seen in Table 3.4. Examples of human rights violations categories with more images are `child labour`, `displaced people`, and

1. Arms	Weapons systems that put civilians at high risk of armed conflict and violence
2. Child Labour	Work that deprives children of their childhood, their potential and their dignity, and that is harmful to physical and mental development
3. Child Marriage	A formal marriage or informal union before age 18. Child marriage is widespread and can lead to a lifetime of disadvantage and deprivation
4. Detention Centres	The right to health and a healthy environment, the right to be free from discrimination and arbitrary detention as critical means of achieving health
5. Disability Rights	People with disabilities experience a range of barriers to education, health care and other basic services, while they are subjected to violence and discrimination
6. Displaced Populations	Abuses against the rights of refugees, asylum seekers, and displaced people (block access to asylum, forcible return of people to places where their lives or freedom would be threatened, and deprive asylum seekers of rights to fair hearings of their refugee claims)
7. Environment	A lack of legal regulation and enforcement of industrial and artisanal mining, large-scale dams, deforestation, domestic water and sanitation systems, and heavily polluting industries can lead to host of human rights violations
8. Out of School	Discrimination of marginalized groups by teachers and other students, long distances to school, formal and informal school fees, and the absence of inclusive education are among the main causes of children staying out of school

Table 3.3: Proposed human rights violation categories with definitions from the Human Rights Archive (HRA) Dataset.

`environment`. Examples of under-sampled categories include `child marriage` and `detention centres`. Figure 3.8a shows the number of images per category, sorted in decreasing order, while Figure 3.8b illustrates the differences among the number of images per violation category for the common 2 violation categories HRUN and HRA.

3.3.4 Visualising HRA

CNNs can be interpreted as continuously transforming the images into a representation in which the classes are separable by a linear classifier. In order to obtain an estimation about the topology of the Human Rights Archive space, we examine the internal features learned by a

	train	val	trainval	test
arms	149	37	186	30
child labour	756	189	945	30
child marriage	69	18	87	30
detention centres	149	37	186	30
disability rights	218	55	273	30
displaced populations	487	122	609	30
environment	326	82	408	30
no violation	162	41	203	30
out of school	123	30	153	30
Total	2439	611	3050	270

Table 3.4: Statistics of the HRA Dataset. The data is divided into two main subsets: training/validation data (`trainval`), and test data (`test`), with the trainval data further divided into suggested training (`train`) and validation (`val`) sets.

Figure 3.9: t-SNE embedding of the HRA Dataset images based on their extracted features. Images that are nearby each other are also close in the CNN representation space, which implies that the CNN 'sees' them as being very similar. Notice that the similarities are more often class-based and semantic rather than pixel and colour-based.

	train	val	trainval	test
child labour	945	25	970	25
non-child labour	945	25	970	25
displaced populations	609	25	634	25
non-displaced populations	609	25	634	25
Total	3108	100	3208	100

Table 3.5: Statistics of the HRA–Binary Dataset. The data is divided into two main subsets: training/validation data (`trainval`), and test data (`test`), with the trainval data further divided into suggested training (`train`) and validation (`val`) sets. Note that each violation category is treated as an independent use case for our subsequent experiments.

CNN using t-SNE (t-distributed Stochastic Neighbour Embedding) [65] visualisation algorithm, by embedding images into two dimensions so that their low-dimensional representation has approximately equal distances as their high-dimensional representation. To produce that visualisation, we feed the HRA set of images through the well studied VGG-16 convolutional-layer CNN architecture [90], where the 4096 dimensional visual features are taken at the output of the second fully-connected layer (*i.e.*, FC7) including the ReLU non-linearity by using Caffe [36] framework. Those features are then plugged into t-SNE in order to project the image features down to 2D. Principal component analysis (PCA) preprocessing is used prior to the t-SNE routine to reduce to 10D to help optimize the t-SNE runtime. We then visualise the corresponding images in a grid as shown in Figure 3.9, which can help us identify various clusters. Every position of the embedding is filled with its nearest neighbour. Note that since the actual embedding is roughly circular, this leads to a visualisation where the corners are a little 'stretched' out and over-represented.

3.3.5 HRA–Binary Dataset

In order to find the main test platform for the evaluation of the extensions of the base method presented in the following chapter, we use HRA to construct a task-specific, two-class subset termed `HRA-Binary Dataset`. We maintain the verified images intact for the two classes with the highest number of samples, `child labour` and `displaced populations`. The HRA–Binary dataset contains 1554 images of human rights violations in total, and the same number of no violation counterparts for training, as well as 200 images collected from the web for testing and validation. Note that each violation category is treated as an independent use case for our subsequent experiments. The dataset is made publicly available for future research [39].

3.4 The Role of Human Rights-Specific Image Datasets

In this chapter, we have discussed all the datasets that were created and will be used throughout this thesis namely the **HRUN**, **HRA**, and **HRA–Binary** datasets. These datasets are used for a range of experiments in this thesis–in Chapter 4 for predicting human rights violations from images, Chapter 5 for comparing object-centric and scene-centric deep image representations, Chapter 6 for recognising displaced people, and Chapter 7 for exploiting emotional traits for human rights violation recognition. We have also highlighted the potential of real-world images in human rights context including the opportunities and challenges they present. The fact that expert verification of image samples is required to ensure their validity, is considered to be of the utmost importance for human rights practitioners. The datasets discussed in this chapter do not represent an exhaustive compilation of image collection procedures that will shape the future of automated analysis of human rights violations, but rather are a starting point to expand our understanding of how an ecosystem of visual context could guide progress on HRVR problems. Next, in Chapter 4 we look at predicting human rights violations from images using the HRUN and HRA datasets.

Chapter 4

Predicting Human Rights Violations from Images: A New Benchmark

---

Categorisation of potential human rights violations plays a crucial role in human rights advocacy and accountability efforts, which is vital for human rights organisations, advocates, journalists, and international institutions. Due to the vast number of human rights violations and the subtle differences among places and events, human rights violation recognition heavily relies on the professional knowledge of humanitarian and human rights experts, meaning it is expensive and time consuming, while the number of researchers or volunteers who are capable of carrying out such work can be limited by language skills, geographic awareness, and cultural knowledge. With the development of deep learning and computer vision techniques, automated HRVR will enable researchers to discover content that may otherwise be concealed by massive volume of visual data. We announce the notion of *human rights violation recognition* as an area of practice within computer vision, by developing the first ever purpose-built human rights violation classification schemes based on deep image representations. To our knowledge, the results presented in this chapter are the first attempt to tackle the HRVR problem using deep visual representations. This chapter can be broken down into two main sections, describing different contributions in classifying human rights violations using deep representations: (i) linear, one-vs-rest, Support Vector Machines (SVMs); (ii) end-to-end learning.

4.1 Introduction

In this chapter, we discuss two different featured-based approaches to create the first ever benchmarks for visual recognition of human rights violations from real-world imagery. We are particularly interested in the domain shift problem of learning such classifiers from images of everyday objects/scenes and applying them to human rights violations, and to what extent this can be rectified with a good feature representation. Our early method is trained, validated and tested on the HRUN dataset (see Section 3.2) utilising a combination of deep representations and a linear Support Vector Machine (SVM). Next, the HRA dataset (see Section 3.3) is used for training, validating and testing our end-to-end approach. This chapter can be broken down into two main sections; the first one focuses on methods to construct *image representations*, *i.e.*, encoding functions ϕ mapping an image I to a vector $\phi(I) \in \mathbb{R}^d$ suitable for analysis with a linear classifier, in Section 4.2. The other section focuses on end-to-end training with the larger HRA dataset. Each section consists of details of the respective models, experiments, and results.

4.2 Combining Deep Representations with a Linear SVM

Our goal is to train a system that recognises different human rights violations from a given input image using the HRUN dataset. One primary issue in this effort is how to find a good representation for instances in such a unique domain. Our deep representations are inspired by the success of the CNN of Krizhevsky *et al.* [54]. As shown in [13, 115], the vector of activities $\phi_{CNN}(I)$ of the penultimate layer of a deep CNN, learnt on a large dataset such as ImageNet [12] or Places [119], can be used as a powerful image descriptor applicable to other datasets. This method, referred to as transfer learning (see Section 2.7), is implemented by taking a pre-trained CNN, replacing the fully-connected layers, and consider the rest of the ConvNet[1] as a fixed feature extractor for the relevant dataset. By 'freezing' the weights of the convolutional layers, the deep ConvNet can still extract general image features such as edges, while the fully connected layers can take this information and use it to classify the data in a way that is applicable to the problem. Here we adopt a single learning framework and experiment with architectures of different complexity exploring their performance-complexity trade-off.

[1] 'ConvNet' is used interchangeably with the term 'CNN'

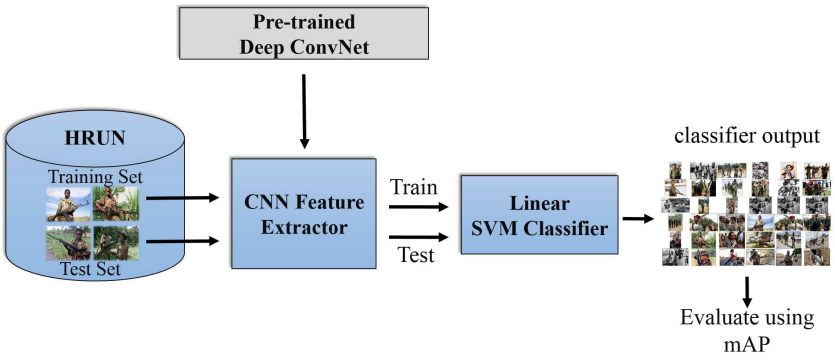


Figure 4.1: Overview of the human rights violation recognition pipeline using a linear SVM classifier. Different deep convolutional models are plugged into the pipeline one at a time, while the training and test samples taken from the HRUN dataset remain fixed. Mean average precision(mAP) metric is used for evaluating the results.

4.2.1 Implementation

The entire pipeline used for combining deep image representations with a linear SVM is depicted in Figure 4.1, and detailed further below. In this pipeline, every block is fixed except the feature extractor as different deep convolutional networks are plugged in, one at a time, to compare their performance utilising the mean average precision (mAP) metric.

Given a training dataset T_r consisting of m human rights violation categories, a test dataset T_s comprising unseen images of the categories given in T_r, and a set of n pre-trained CNN architectures (C_1,...C_n), the pipeline operates as follows: The training dataset T_r is used as input to the first CNN architecture C_1. The output of C_1, as described above, is then utilised to train m SVM classifiers. A one-vs-rest SVM classifier for each class is learnt and evaluated independently and the performance is measured as mAP across all classes. The training and testing procedures are then repeated after replacing C_1 with the second CNN architecture C_2 to evaluate the performance of the human rights violation recognition pipeline. For a set of n pre-trained CNN architectures, the training and testing processes are repeated n times. Since the entire pipeline is fixed (including the training and test datasets, learning procedure and evaluation protocol) for all n CNN architectures, the differences in the performance of the classification pipeline can be attributed to the specific CNN architectures used.

For comparison, 10 different deep CNN architectures were selected, grouped by the common paper which they were first made public: a) 50-layer ResNet, 101-layer ResNet and 152-layer ResNet presented in [31]; b) 22-layer GoogLeNet [95]; c) 16-layer VGG-Net and 19-layer VGG-Net introduced in [90] ; d) 8-layer VGG-S, 8-layer VGG-M and 8-layer VGG-F displayed in [6]; and e) 8-layer Places [120]. To ensure a fair comparison, all the standardised CNN

model	dimensional representation	mAP	child labour	child soldiers	police violence	refugees
ResNet50	100K	42.59	41.12	43.69	43.81	41.73
ResNet101	100K	42.07	40.48	44.78	42.56	40.48
ResNet152	100K	45.80	44.27	44.11	48.08	46.73
GoogLeNet	50K	48.62	42.72	40.71	61.91	49.16
VGG16	4K	77.46	70.79	**77.71**	83.46	77.87
VGG19	4K	47.01	31.69	50.98	73.79	31.57
VGG M	4K	67.93	59.52	62.96	81.45	67.80
VGG S	4K	**78.19**	**80.17**	64.46	87.46	**80.68**
VGG F	4K	64.15	45.42	63.20	84.78	63.21
Places	4K	68.59	55.67	65.60	**93.17**	59.92

Table 4.1: Human rights violation classification results on the test set of HRUN using a 70/30 split for training and testing images. Mean average precision (mAP) accuracy for different CNNs. Bold font highlights the leading mAP result for every experiment.

models[2] used in our experiments are based on the open source Caffe framework [36] and are pre-trained on 1000 ImageNet [12] classes with the exception of Places CNN which was trained on 205 scenes categories of Places database [120]. For the majority of the networks, the dimensionality of the last hidden layer (FC7) leads to a 4096×1 dimensional image representation. Since GoogLeNet and ResNet architectures do not utilise fully connected layers at the end of their networks, the last hidden layers before average pooling at the top of the ConvNet are exploited with $7 \times 7 \times 1024$ and $7 \times 7 \times 2048$ feature maps respectively, to counterbalance the behaviour of the pool layers, which provide downsampling regarding the spatial dimensions of the input.

The evaluation process is divided into two different scenarios, each one making use of an explicit split of images between the training and testing samples of the pipeline. For the first scenario, a split of 70/30 was utilised, while for the second scenario the split was adjusted to 50/50 for training and testing images respectively. Additionally, three distinct series of tests were conducted for each scenario, each and every one assembled with a completely arbitrary shift of the entire image set for every category of the HRUN dataset. This approach ensures an unbiased comparison with a rather limited dataset like HRUN at present. The compound results of all three tests are given in Table 4.1 and Table 4.2 and analysed below.

model	dimensional representation	mAP	child labour	child soldiers	police violence	refugees
ResNet50	100K	70.94	73.15	68.07	70.44	72.09
ResNet101	100K	68.46	69.50	66.90	68.34	69.09
ResNet152	100K	76.20	80.60	73.07	72.00	79.12
GoogLeNet	50K	55.92	41.48	60.21	55.52	66.48
VGG16	4K	84.79	79.15	87.94	89.47	82.59
VGG19	4K	60.39	35.72	72.67	83.10	50.08
VGG M	4K	78.94	68.71	82.32	89.99	74.74
VGG S	4K	**88.10**	**84.84**	88.14	91.92	**87.50**
VGG F	4K	73.46	53.57	78.78	90.41	71.08
Places	4K	81.40	62.04	**89.97**	**95.70**	77.90

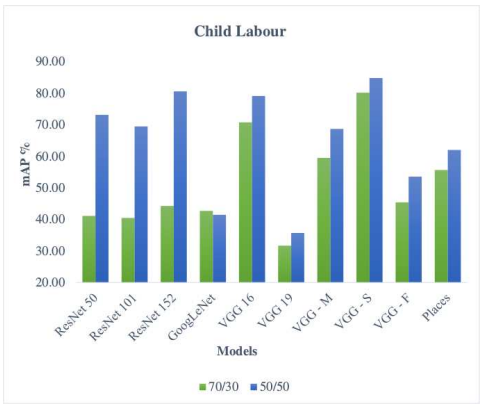
Table 4.2: Human rights violation classification results on the test set of HRUN using a 50/50 split for training and testing images. Mean average precision (mAP) accuracy for different CNNs. Bold font highlights the leading mAP result for every experiment.

4.2.2 Results and Discussion

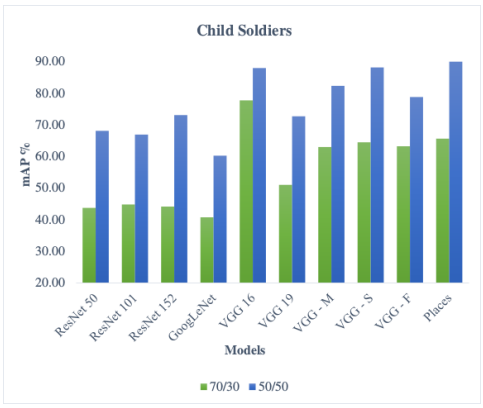
It is evident from Table 4.1 and Table 4.2 that the *Slow* CNN architecture performs the best for the child labour category for both scenarios. VGG with 16 layers performs the best in the case of child soldiers within scenario 1, while scenario's 2 best performing architecture is *Places* with *VGG16* coming genuinely close. Places was also the best performing architecture for the category of police violence for the two scenarios. Lastly, regarding the refugees category, the Slow version of *VGG* was the dominant architecture for both scenarios. The best performing architectures can achieve up to 88.10% mean average precision when recognising human rights violations. On the other hand, some of the regularly top performing deep ConvNets, such as *GoogLeNet* and *ResNet*, fell short for this particular task compared to the others. Such weaker performance occurs primarily because of the limited dataset size, whereby learning millions of parameters of those very deep convolutional networks is usually impractical and may lead to over-fitting. Another interpretation could be due to the inadequate structure of the image representation deduced from the last hidden layer before average pooling compared to the FC7 layer of the others.

Surprisingly, a 50/50 split of images in the course of scenario 2 provides a considerable boost in performance of the HRVR pipeline as compared to the first scenario when a split of 70/30 was employed for training and testing images respectively. Figure 4.2 depicts the effect of two varying training data sizes (scenario 1 vs scenario 2) on the performance of different deep convolutional networks. Remarkably, scenario 2 where the 50/50 split was applied, accomplishes a notable improvement on mean average precision which spans from 4.03% up to

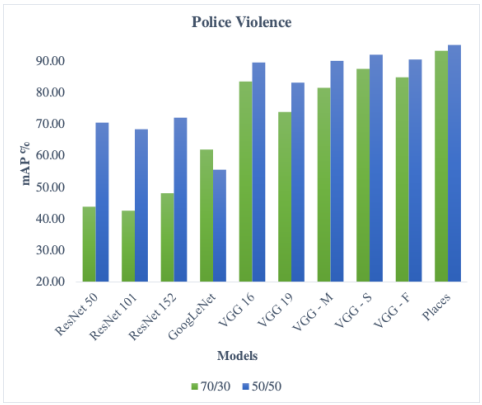
[2]Available at https://github.com/GKalliatakis/Human-Rights-UNderstanding-CNNs



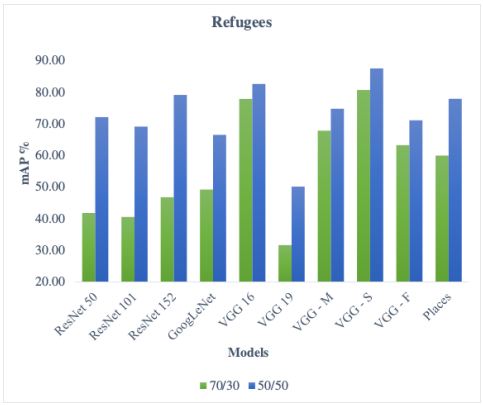
(a) child labour



(b) child soldiers



(c) police violence



(d) refugees

Figure 4.2: Comparison of deep convolutional networks performance, with reference to mAP, for the two different scenarios appearing in our experiments. The number on the left side of the slash denotes the training proportion of images, while the name on the right implies the testing percentage.

36.33% across all four HRUN categories which were tested. Only on two occasions scenario 2 was outperformed by scenario 1, both of them while *GoogLeNet* was selected for the categories of 'child labour' and 'police violence'. This observation strengthens the point discussed above with respect to over-fitting.

Deep CNNs will generally perform best when their capacity is appropriate in regard to the true complexity of the task they need to perform and the amount of training data they are provided with. Models with insufficient capacity are unable to solve complex tasks, while models with high capacity can solve complex tasks, but when their capacity is higher than needed to solve the present task they may overfit. Overfitting refers to a model that represents the 'training data' extremely accurately. Overfitting occurs when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. An indication of overfitting may be seen in the classification accuracy on the

training data. If the training accuracy is exceeding the test accuracy, it means that our model is learning details and noises of training data and specifically working of training data. In the next section, we will try to prevent our models from learning misleading or irrelevant patterns found in the training data, by applying some regularization techniques in practice to improve the human-rights-violations-classification model and create a new benchmark.

4.3 End-to-End Image Classification of Human Rights Violations

In this section, we use features extracted from networks trained on objects and scenes as a generic image representation to tackle the unique task of human rights violation recognition similar to Section 4.2. However, the classifiers examined here follow an end-to-end approach, while they are trained and fine-tuned on features from the training set of the latest and larger HRA dataset. A typical structure of an end-to-end system for image classification using representation learning methods is depicted in Figure 4.3.

4.3.1 Implementation

Given the impressive classification performance of the deep convolutional neural networks, we choose three popular object-centric CNN architectures, ResNet50 [31], VGG 16 convolutional-layer CNN, and VGG 19 convolutional-layer CNN [90], then fine-tune them on HRA to create baseline CNN models. Additionally, given the nature of the task at hand, we further fine-tuned a scene-centric CNN architecture, VGG16-Places365 [119] and compared it with the object-centric CNNs for human rights violation recognition. We also trained a small CNN on the HRA training samples from scratch to set a baseline for what can be achieved. The baseline model is a simple stack of 3 convolution layers with a ReLU activation, followed by max-pooling layers. This is very similar to the architecture that LeCun *et al.* [56] advocated in the 1990s for image classification (with the exception of ReLU). Finally, we employed the above CNNs as fixed feature extractors by removing their classification block and computing a vector for every image in the HRA dataset, before training a nearest neighbour classifier with those extracted features. All the HRA-CNNs [3] presented here were trained using the Keras package [8] on Nvidia GPU Tesla K80.

The baseline CNN contains 3.2 million parameters, while the other selected CNN architectures contain 138 million parameters for VGG16, 143 million parameters for VGG19 and 26

[3] Available at https://github.com/GKalliatakis/Human-Rights-Archive-CNNs

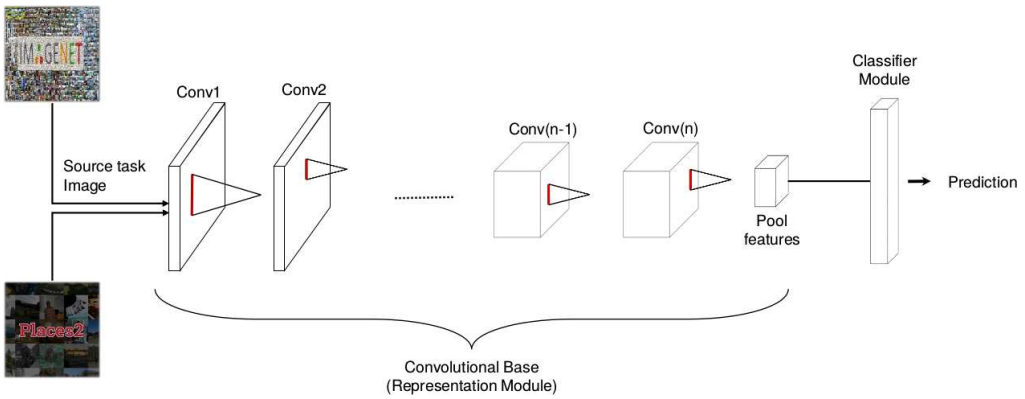


Figure 4.3: Typical structure of an end-to-end system for image classification using representation learning methods.

million parameters for ResNet50. VGG16-Places365 and VGG16 have exactly the same network architecture, but they are trained on scene-centric data and object-centric data respectively. Directly learning so many parameters from only a few thousand training images is problematic.

4.3.2 Transferring CNN weights

A conventional approach to enable training of very deep networks on relative small datasets is to use a model that has already been trained on a very large dataset, and then use the CNN as an initialization for the task of interest. This method, referred to as 'transfer learning' [76, 13, 115] injects knowledge from other tasks by deploying weights and parameters from a pre-trained network to the new one [45] and has become a commonly used method to learn task-specific features. The key idea is that the internal layers of the CNN can act as a generic extractor of image representations, which can be pre-trained on one large dataset, the *source task*, and then re-used on other *target tasks* [74]. Considering the size of our dataset, a reasonable approach is to try and reduce the number of free parameters. In order to achieve this, the first filter stages can be trained in advance on different tasks–object or scene recognition–and held fixed during training on HRVR. By freezing the earlier layers (preventing the weights from getting updated during training), overfitting can be avoided. We initialize the feature extraction modules using pre-trained models from two different large scale datasets, ImageNet [12] and Places[119].

ImageNet is an object-centric dataset which contains images of generic objects including *person*, and therefore is a good option for understanding the contents of the image region comprising the target person. On the contrary, Places is a scene-centric dataset specifically created for high level visual understanding tasks such as recognising scene categories. Hence, pretraining the image feature extraction model using this dataset ensures global, high level contextual support. We treat the target task of human rights violation recognition as a *single-*

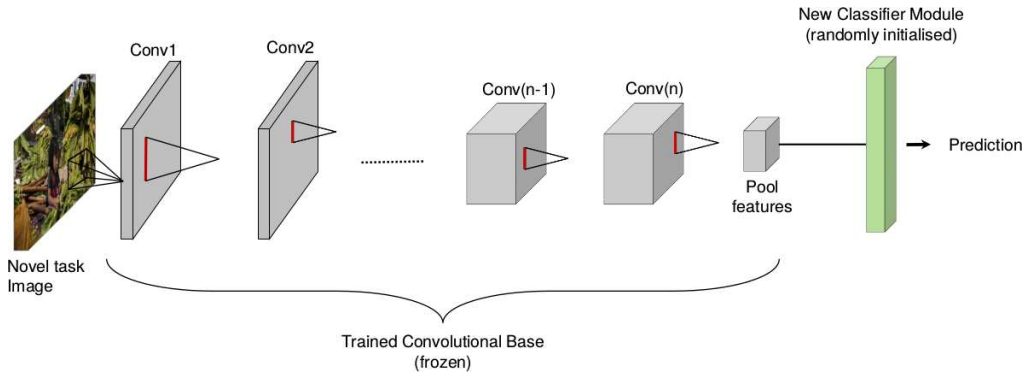


Figure 4.4: Network architecture used for high-level feature extraction with the HRA Dataset. Pre-trained parameters of the internal layers of the networks are transferred to the target task. To compensate for the different nature of the source and target data we add a randomly initialised adaptation layer (fully connected layer) and train them on the labelled data of the target task.

label, *multi-class classification* problem. We design a network that will output scores for the eight target categories of the HRA dataset or `no violation` if none of the categories are present in the image.

Feature extraction

Transfer is achieved in two phases. First, we start by using the representations learned by a previous network in order to extract interesting features from new samples. Feature extraction consists of taking the convolutional base of a pre-trained network, running the new data of HRA through it and training a new, randomly initialised classifier on top of the semantic image output vector $\mathbf{Y}_{out}$, as illustrated in Figure 4.4. We intentionally utilise only the convolutional base and not the densely-connected classifier of the original network. The reason is that the representations learned by the convolutional base are likely to be more generic and therefore more reusable. On the other hand, the representations learned by the classifier will inevitably be specific to the set of classes on which the model was trained. Additionally, representations found in densely connected layers no longer contain any spatial information, these layers eliminate the notion of space, whereas the object location is still described by convolutional feature maps. Note that $\mathbf{Y}_{out}$ is obtained as a complex non-linear function of potentially all input pixels and captures the high-level configurations of objects or scenes. Note that in our experiments, the operation applied on the frozen convolutional base just before the new classifier can be either a global average/max pooling operation for spatial data or simply a flattening layer. The FC_{HRA} layer compute $\mathbf{Y}_{HRA} = \sigma(\mathbf{W}_{HRA}\mathbf{Y}_{out} + \mathbf{B}_{HRA})$, where $\mathbf{W}$, $\mathbf{B}$ are the trainable parameters. In all our experiments, the last convolutional layer of the pre-trained base have sizes of $7 \times 7 \times 512$.

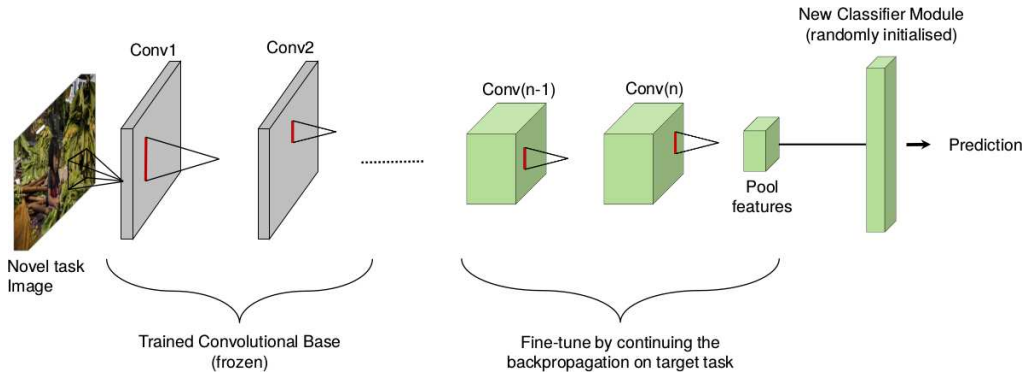


Figure 4.5: Network architecture used for fine-tuning with the HRA Dataset. It marginally alters the more abstract representations of the model being utilised, in order to make them more relevant for the problem at hand.

Fine-tuning

The second phase required for transferring CNN weights, complementary to feature extraction, is fine-tuning. It consists of unfreezing few of the top layers of a previously frozen convolutional base for feature extraction, and jointly training both the newly added fully-connected classifier and these top layers as illustrated in Figure 4.5. It is only beneficial to fine-tune the top layers of the convolutional base once the classifier on top has already been trained (see Figure 4.4). If the classifier is not already trained, then the error signal propagating through the network during training will be too large, and the representations previously learned by the layers being fine-tuned will be destroyed. We choose to fine-tune only the last two convolutional layers for two reasons. First, earlier layers in the convolutional base encode more-generic, reusable features, whereas layers higher up the network encode task-specific features. It is more useful to fine-tune the task-specific features, because these are the ones that need to be repurposed on our new problem. There would be fast-decreasing returns in fine-tuning lower layers. Second, the more parameters we are training, the more we are at risk of overfitting. The convolutional base has million of parameters, so it would not be sensible to attempt training it on our small dataset.

For all of our experiments, we use the HRA dataset (Section 3.3) exclusively for the training process, while we obtain other representative images for each category from the Internet in order to compose the test set, producing a total of 270 valid images. Thus, we eliminate the presence of bias in our experiments while our models are tested in the wild with real-world images. For the purposes of our experiments, the data is divided into two main subsets: training/validation data (trainval), and test data (test) as illustrated in Table 3.4. To compensate for the imbalanced classes in HRA, we utilise cost-sensitive training to weight the loss function during training by an amount proportional to how under-represented each class is. This is useful to tell the model

to 'pay more attention' to samples from an under-represented class. The maximum number of epochs was set to 40 iterations for each epoch and a learning rate of 0.0001, using the stochastic gradient descent (SGD) optimizer for cross-entropy minimization. The parameters were chosen empirically by analysing the training loss.

4.3.3 Performance Metrics

An important consideration in deep learning is the choice of which performance metric to use. Several different performance metrics may be used to measure the effectiveness of a complete application that includes deep learning components. These performance metrics are usually different from the cost function used to train the model. For tasks such as image classification we often measure the *accuracy* of the model. Accuracy is just the proportion of examples for which the model produces the correct output, while we can obtain equivalent information by measuring the *error rate*, the proportion of examples for which the model produces an incorrect output. Usually we are interested in how well a deep learning model performs on data that has not seen before, using the *top-1 accuracy*, which is when the model's prediction with highest probability is exactly the correct answer (ground truth label).

In some applications it is possible for the system to reject a prediction. This is suitable when the model can estimate how confident it should be about a decision, particularly if a wrong decision can be harmful and if a human operator is supposed to take over. Human rights violations present an example of this situation. The value of the recognition system deteriorates considerably if the prediction is inaccurate. Therefore, it is important to point out images that potentially depict human rights violations only if the confidence of the prediction is above a threshold. Of course, an automated system is only useful if it is able to effectively reduce the amount of photos that a human rights investigator must process. A realistic performance metric to use in scenarios in which deep learning models can often produce no response is *coverage*. This metric qualifies the fraction of examples for which the system is able to produce a response/prediction. It is possible to trade coverage for accuracy. For example, one can always obtain 100% accuracy by refusing to process any example, but this reduces the coverage to 0%. In order to create a criterion from which an overall optimum can be easily envisaged when comparing different models and settings, we also compute a weighted sum of the two performance metrics, $weighted\_sum = 0.25(top\text{-}1\ acc. + coverage)$.

4.3.4 Results

After fine-tuning the various CNNs, we used the final output layer of each network to classify the test set images of the HRA dataset. The classification results, using the cost-sensitive

	Operation on Conv. Base	Top-1 acc.	Coverage	Weighted Sum	Train Params.
Baseline-CNN		12.59%	61%	18.39	3,240,553
VGG16	avg-pool	34.44%	45%	19.86	4,853,257
VGG19		**35.18%**	42%	19.29	4,853,257
ResNet50		25.55%	55%	20.13	4,992,521
VGG16-places365		30.00%	32%	15.5	4,853,257
VGG16	flatten	31.85%	55%	21.71	8,784,905
VGG19		31.11%	50%	20.27	8,784,905
ResNet50		30.00%	44%	18.5	4,992,521
VGG16-places365		28.51%	52%	20.12	8,784,905
VGG16	max-pool	28.14%	**64%**	**23.03**	4,853,257
VGG19		29.62%	61%	22.65	4,853,257
ResNet50		25.55%	61%	21.63	4,992,521
VGG16-places365		26.66%	51%	19.41	4,853,257
VGG16 L2		22.59%	37%	14.89	-
VGG19 L2		24.44%	42%	16.61	-
ResNet50 L2		11.11%	18%	7.27	-
VGG16-places365 L2		18.51%	34%	13.12	-

Table 4.3: Classification accuracy on the test set of HRA using our proposed fine-tuned CNNs alongside two other baseline models, a CNN trained from scratch (first row) and a nearest neighbour classifier for the extracted features (last four rows). 'Weighted Sum' refers to the derived criterion for finding the overall optimum. Bold font highlights the dominant performance across the same metric.

training, for *top-1 accuracy* and *coverage* are listed in Table 4.3. For the sake of completeness, we also provide classification results without weighting the loss function during training as illustrated in Table 4.4 and with real time data augmentation during training in Table 4.5. Not weighting the loss function during training results in a significant drop of 4.23 points in the best reported weighted sum. This rather weak score suggest that our initial intuition of training imbalanced classes equitably by increasing the importance of the under-represented classes has indeed a positive effect on both accuracy and coverage. Although on paper applying a number of random transformations in order to augment our training samples will help the models generalise better, as revealed by lower weighted sum scores, data augmentation does not improve the accuracy and coverage of the models for most of the cases. Note that for all the remaining experiments presented in this paper, results concerning only the superior cost-sensitive training are indicated. Given that a system capable of recognising human rights violations from visual content is only useful if it achieves high coverage, it was important to set a strong coverage requirement for this task. Specifically, the network refuses to classify

	Operation on Conv. Base	Top-1 acc.	Coverage	Weighted Sum	Train Params.
Baseline-CNN		15.55%	34%	12.38	3,240,553
VGG16	avg-pool	25.92%	23%	12.23	4,853,257
VGG19		24.07%	32%	14.01	4,853,257
ResNet50		17.40%	2%	4.85	4,992,521
VGG16-places365		26.66%	16%	10.66	4,853,257
VGG16	flatten	27.40%	41%	17.1	8,784,905
VGG19		28.88%	41%	17.47	8,784,905
ResNet50		18.50%	4%	5.62	4,992,521
VGG16-places365		25.55%	49%	18.63	8,784,905
VGG16	max-pool	**28.51%**	38%	16.62	4,853,257
VGG19		22.22%	**53%**	**18.80**	4,853,257
ResNet50		10.74%	2%	3.18	4,992,521
VGG16-places365		25.55%	40%	16.38	4,853,257

Table 4.4: Classification accuracy and coverage on the test set of HRA using our proposed fine-tuned CNNs without weighting the loss function during training. 'Weighted Sum' refers to the derived criterion for finding the overall optimum. Bold font highlights the dominant performance across the same metric.

	Operation on Conv. Base	Top-1 acc.	Coverage	Weighted Sum	Train Params.
Baseline-CNN		13.70%	17%	7.67	3,240,553
VGG16	avg-pool	33.70%	37%	17.67	4,853,257
VGG19		32.59%	33%	16.39	4,853,257
ResNet50		24.81%	59%	20.95	4,992,521
VGG16-places365		25.18%	26%	12.79	4,853,257
VGG16	flatten	**34.07%**	34%	17.01	8,784,905
VGG19		**34.07%**	27%	15.26	8,784,905
ResNet50		24.44%	**64%**	**22.11**	4,992,521
VGG16-places365		26.29%	37%	15.82	8,784,905
VGG16	max-pool	32.22%	43%	18.80	4,853,257
VGG19		27.40%	60%	21.85	4,853,257
ResNet50		22.96%	54%	19.24	4,992,521
VGG16-places365		24.07%	43%	16.76	4,853,257

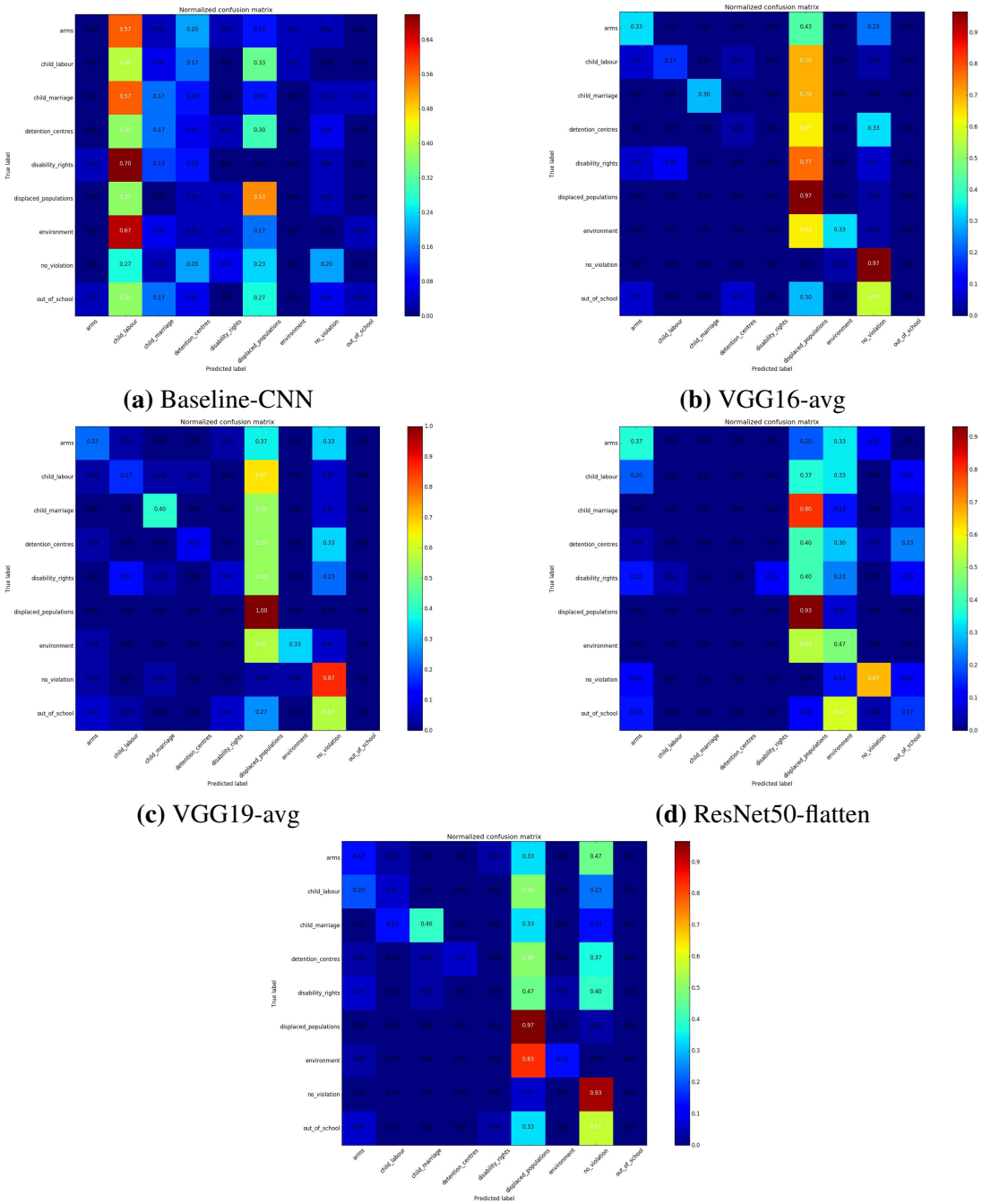
Table 4.5: Classification accuracy and coverage on the test set of HRA using our proposed fine-tuned CNNs and real-time data augmentation during training. 'Weighted Sum' refers to the derived criterion for finding the overall optimum. Bold font highlights the dominant performance across the same metric.

Figure 4.6: The predictions given by the best performing HRA-VGG19 for the images from the HRA test set. The ground-truth label (GT) and the top 3 predictions are shown. The number beside each label indicates the prediction confidence.

an input x, whenever the probability of the output sequence $p(y|x) < t$ for some confidence threshold t. For all the experiments in this thesis, we set the confidence threshold at 0.85 in order to report the coverage performance.

Figure 4.6 shows the responses to examples predicted by the best performing HRA-CNN, VGG19 with max pooling. Broadly, we can identify one type of misclassification given the current label attribution of HRA: images depicting the evidence which are responsible for a particular situation and not the actual action, such as schools being targeted by armed attacks. Future development of the HRA database, will explore to assign multi-ground truth labels or free-form sentences to images to better capture the richness of visual descriptions of human rights violations. Figure 4.7 illustrates the normalised confusion matrices of the best performing CNNs. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabelled by the classifier. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions. This kind of normalisation can be beneficial in case of class imbalance to have a more visual interpretation of which class is being misclassified. These results indicate that predictions relying solely on object-based information are likely to misinterpret visual samples that belong to the class of disability rights as displaced populations. Other examples where the CNNs make mistakes are: predicting detention centres as displaced populations, and out of school as no violation. This is not surprising because these pairs share similar properties, *e.g.* numerous people gathered at one place.

We can see from Table 4.3 that both VGG architectures surpass the scene-centric architecture of VGG16-Places365 by a significant margin of at least 4.44% for top-1 accuracy and 10% for coverage–using the average pooling which is their best performing operation–even though



(a) Baseline-CNN

(b) VGG16-avg

(c) VGG19-avg

(d) ResNet50-flatten

(e) VGG16-places365-avg

Figure 4.7: Normalised confusion matrices of the best performing HRA-CNNs. A row represents an instance of the actual class, whereas a column represents an instance of the predicted class. The values of the diagonal elements represent the degree of correctly predicted classes. Results indicate that predictions relying solely on object-based information are likely to misinterpret visual samples that belong to the class of disability rights as displaced populations.

(a) input image **(b)** object-centric CNN **(c)** scene-centric CNN

Figure 4.8: Given an input image, we visualise the class-discriminative regions of different CNNs using Grad-CAM [85] for the output classes `child labour` and `child marriage`. Note that the object-centric model (b) focuses on the tool used by a child or the clothing, while the scene-centric model (c) focuses on the vegetation or a sign in the background.

the number of trainable parameters remains exactly the same. On the other hand, VGG16-Places365 outperforms the object-centric ResNet50 for two of the operations applied on the frozen convolutional base. We have also tried to change the number of layers which were fine-tuned in our training set-up. Increasing the number of layers to three results in about 7% drop in classification performance.

4.3.5 Interpreting the Deep Neural Networks

In order to interpret which parts of a given image led a CNN to its final prediction, we produce heatmaps of 'class activation'. Class Activation Mapping (CAM) [118] and its generalisation Gradient-weighted Class Activation Mapping (Grad-CAM) [85] visualise the linear activations of a late layer's activations with respect to the class considered. To generate Grad-CAM visual explanations, we followed the approach presented in [85]. An image is fed into the fine-tuned network and the output feature maps of the last convolutional layer are extracted. Convolutional features are capable of retaining spatial information compared to fully-connected layers where that information is lost. The gradient of the score associated with a specific output class is computed, with respect to the extracted feature maps of the last convolutional layer. Then, the gradients are global-average-pooled to obtain the importance weights. Finally, the Grad-CAM is obtained by performing a weighted combination of forward activation maps followed by

a ReLU. Figure 4.8 shows an example of Grad-CAMs for the output class of output classes `child labour` and `child marriage`.

4.4 Summary

In this chapter we have examined the transfer learning problem of classifying human rights violations using classifiers learnt from images of everyday objects/scene using deep features. First, we have observed that a combination of deep representations and a linear SVM over the small-scale HRUN image dataset suffers from overfitting. To prevent a model from learning misleading or irrelevant patterns found in the training data, the best solution is to get more training data. A model trained on more data will naturally generalize better. The reason is that, as we add more data, the model becomes unable to overfit all the samples, and is forced to generalize to make progress.

Following this observation, we fine-tuned end-to-end models for object classification (object-centric) and scene classification (scene-centric), over the training set of the larger HRA image dataset. Unsurprisingly, each object-centric and scene-centric CNN has different strengths and weaknesses, as shown in Table 4.3 and Figure 4.8. Motivated by this observation, we then look to capture robust representations from the perspectives of object and scene. In Chapter 5, our aim is to develop an ensemble of object-centric and scene-centric CNNs, investigate different network architectures on the task of HRVR and also explore their complementarity by fusing them.

Chapter 5

Objects and Scenes: Combining Features for Human Rights Violation Recognition

---

One would think that the meaningful parts of an image depicting a potential violation against human rights are the tools, weapons, and humans. However, those are simply functional parts, with words associated with them; the object parts that are important for visual recognition might be different from their semantic counterparts, making it difficult to evaluate how efficient a representation is. In fact, the strong internal structure of objects makes the definition of what is considered a useful part inadequately constrained: an algorithm can find different and arbitrary part configurations, all giving similar recognition performance. Learning to classify human rights violations (*i.e.*, labelling an image as potentially being `child labour`, `no violation`, `displaced populations`, *etc.*) using features extracted from a Places-CNN gives the opportunity to study the internal representations learned by a CNN on a task other than object recognition. Scene categories are defined by the objects they accommodate and by the spatial configuration of those objects. For instance, the meaningful parts of a dinning room are the table, the chairs, as well as the dinnerware set. Therefore, objects represent a distributed code for scenes (*i.e.*, object classes are shared across different scene categories). The main contribution of this chapter is to investigate whether image features from pretrained ImageNet-CNN and Places-CNN can complement each other for predicting human rights violations and how the differences between these networks can impact the classification performance.

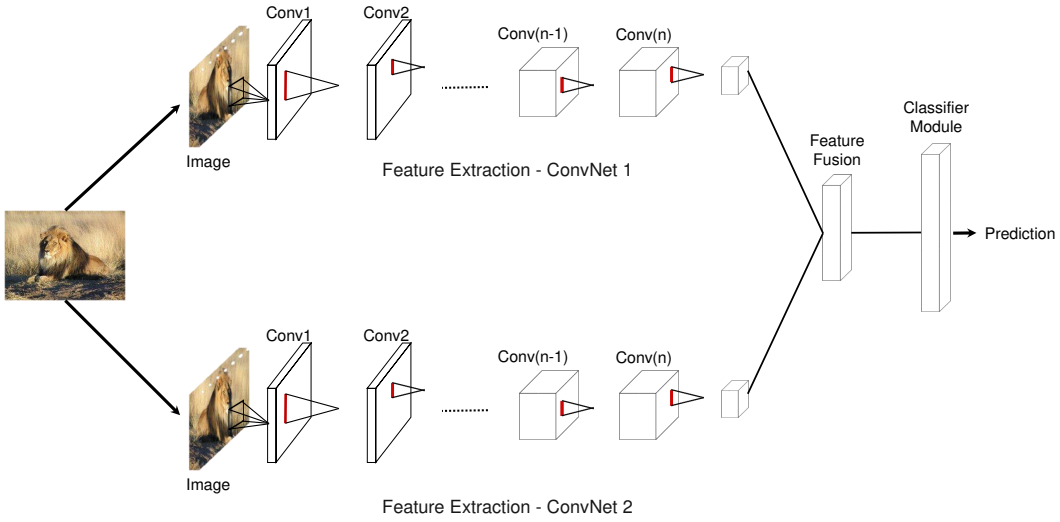


Figure 5.1: Illustration of a typical high-level CNN *early* feature fusion and image classification workflow. The model consist of two feature extraction modules (may originate from different CNN architectures), a fusion network, and a classifier module for making predictions.

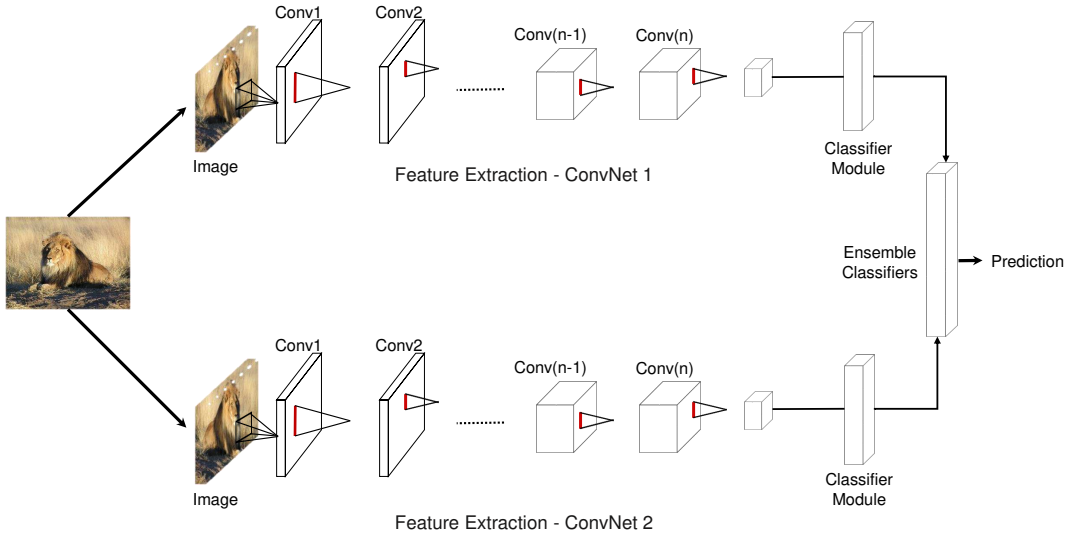


Figure 5.2: Illustration of a typical high-level CNN *late* feature fusion and image classification workflow. The model consist of two feature extraction modules (may originate from different CNN architectures), and a module to ensemble classifiers before making predictions.

5.1 Introduction

Information fusion can be a crucial component in image classification schemes, where increasing the overall accuracy of the system is regarded as one of its most integral aspects. Merging different information is not only meaningful because of the accuracy improvement it

might offer in a system, but also for allowing the system to be more robust against changing dynamics. Since scenes are composed in part of objects, accurate recognition of human rights violations requires knowledge about both. By visualising the class-discriminative regions of object-centric-CNNs and scene-centric-CNNs previously (Figure 4.8) we have found that each model focuses on different aspects of the image in order to classify it. Inspired by this observation we want to investigate whether feature fusion–accommodating features coming from different sources into a single representation–which has resulted in increased performances in recent works [105, 109], would have similar effects on predicting human rights violations.

Single modality CNN features do not capture higher order feature interactions that are pivotal in many visual classification and recognition tasks. Thus, an applicable solution is to fuse multi-modal CNN features. Suppose, the high-level features from different ConvNets have complementary cues, then the fusion layer must learn the correspondence between these features in order to be able to discriminate between the different classes. As indicated by [91, 18], feature fusion approaches can be grouped into two main categories: *early* and *late* fusion. Typical workflows of early and late fusion schemes are depicted in Figure 5.1 and Figure 5.2 respectively.

5.2 Proposed Fusion Schemes

5.2.1 Early Fusion

Suppose we are given two CNNs. Let feature set $F = \{f_1, f_2\}$ be extracted features of the last convolutional layer from each network, where each feature is a high-dimensional feature vector represented with $f_i \in \mathbb{R}^{d_i}$. Every distinct feature may have different cardinality according to the particular CNN architecture, such that $d_i = \{d_1, d_2\}$. Then the feature fusion function ϕ can be defined as the mapping operator on F such that $\phi(F) \mapsto R^d$.

The first strategy we exploit in our proposed early fusion scheme is the *concatenation* method, where discrete feature vectors of different sources are concatenated into one super-vector $f_f = \{f_1, f_2\}$ which will represent the final image feature. The final vector size is the summation of all feature dimensions $d = \sum_{i=1}^{n} d_i$.

The second fusion strategy we employ is *averaging*, also known as *sum pooling* in the context of neural networks. In this strategy, the feature set F is averaged in order to form the final image descriptor $f_f = \frac{1}{n}\sum_{i=1}^{n} f_i$. All features in F should either have the same cardinality or the feature dimensions must be normalised prior to the fusion operation.

The last fusion strategy utilised is *maximum pooling*. It involves the same preprocessing in terms of the final feature cardinality, however it varies in the way features are merged.

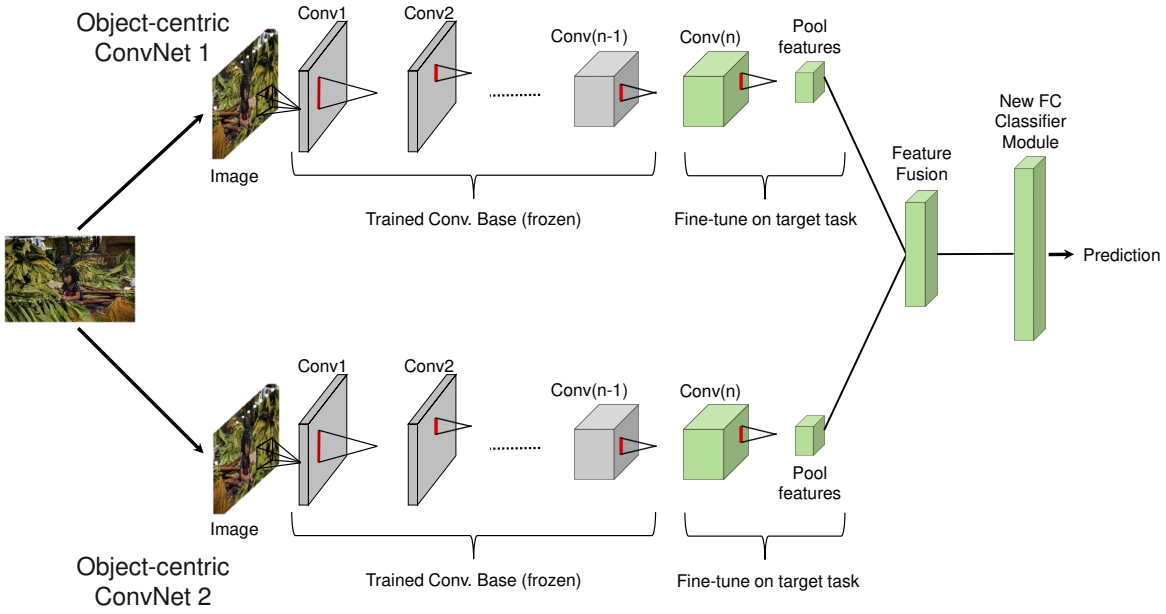


Figure 5.3: Illustration of our proposed object-centric high-level CNN *early* feature fusion and image classification system. The model consist of two different object-centric feature extraction modules, a fusion network, and a classifier module for making predictions. Grey colour indicates the frozen convolutional layers, while the light green colour indicates the layers that will be differently randomly initialised.

Maximum pooling selects the highest value from the corresponding features instead of taking the average of all features elements' in sum pooling. If the final feature representation is $f_f \mapsto R^d$, then max pooling selects each member of f_f as $f_f^i = \max_{i=1}^{d}(f_1^i, f_2^i)$.

5.2.2 Late Fusion

In late fusion, also known as decision fusion, extracted features are processed separately and only the results are combined. The output of each classifier is combined to arrive at a final result. Given multiple classifiers trained with different features, late fusion tries to combine the prediction scores of all classifiers (the prediction score of each sample generated by a classifier indicates the confidence of classifying the sample as positive). Such fusion method is expected to assign positive samples higher fusion scores than the negative ones so that the overall performance can be improved. Although very simple, this method has proved to be effective in improving performance of each individual classifier and also produces highly comparative results to multi-feature early fusion methods [91, 113]. Our late fusion scheme consists of pooling together the predictions of a set of different fine-tuned end-to-end models,

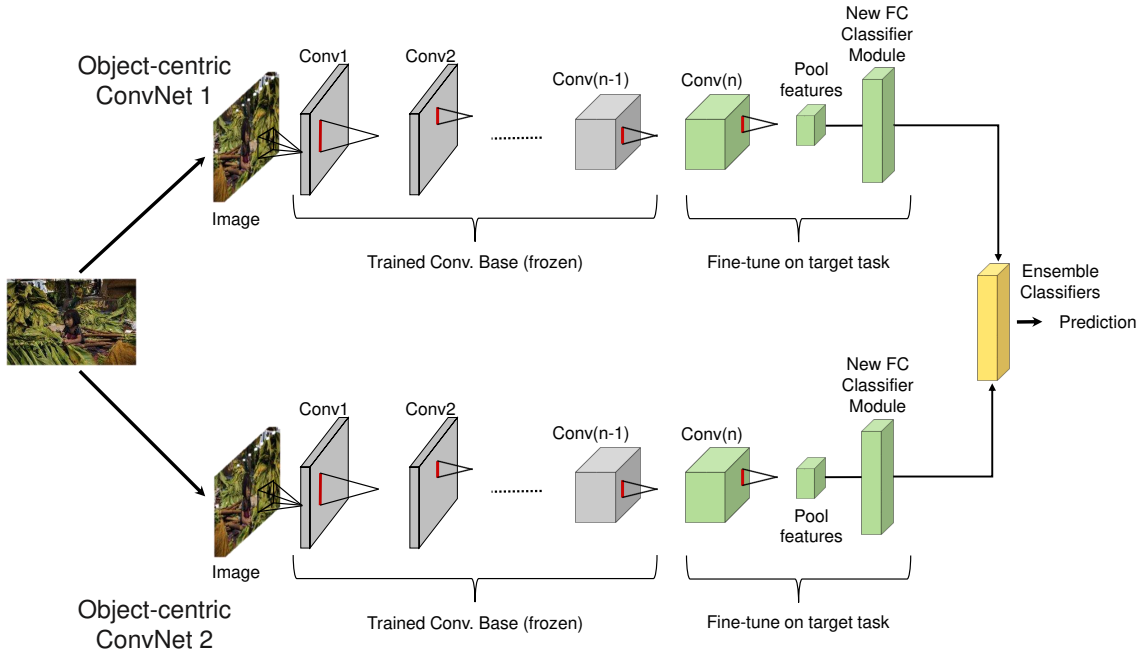


Figure 5.4: Illustration of our proposed object-centric high-level CNN *late* feature fusion and image classification system. The model consist of two different object-centric feature extraction modules, and a module to ensemble classifiers before making predictions. Grey colour indicates the frozen convolutional layers, light green colour indicates the layers that will be randomly initialised, and light yellow colour indicates the ensemble of classifiers used to make the final prediction.

to produce more accurate predictions. This kind of assemblage relies on the assumption that independently trained models are focusing on slightly different aspects of the data to make their predictions. The easiest way to pool the predictions of a set of classifiers is to average their predictions at inference time as illustrated by Figure 5.2.

5.3 Object-Centric Feature Fusion

After evaluating feature extraction and transfer learning schemes on the test set of HRA, we turn our attention to the problem of combining those features for the same task. First, we start by transferring CNN weights as described previously, this time combining the outputs of the last convolutional layers of two different object-centric CNNs before randomly initialising a new fully-connected classifier. Note that in this approach only the last convolutional layer of each network is fine-tuned in order to keep equal number of trainable parameters with the previous set-up introduced in Chapter 4. The processing pipeline of our early fusion scheme for predicting human rights violations is depicted in Figure 5.3. We also employ a

	Operation on Conv. Base	Fusion Strategy	Top-1 acc.	Coverage	Weighted Sum	Train Params.
VGG16 + ResNet50	avg-pool	average	32.59%	7%	9.89	2,494,473
		concatenate	24.81%	16%	10.20	2,625,545
		maximum	32.59%	17%	12.39	2,494,473
	flatten	average	28.14%	40%	17.03	8,785,929
		concatenate	28.14%	51%	19.78	15,208,457
		maximum	30.00%	51%	20.25	8,785,929
	max-pool	average	26.29%	34%	15.07	2,494,473
		concatenate	30.37%	41%	17.84	2,625,545
		maximum	22.59%	32%	13.64	2,494,473
VGG19 + ResNet50	avg-pool	average	35.55%	16%	12.88	2,494,473
		concatenate	**36.29%**	20%	14.07	2,625,545
		maximum	29.62%	22%	12.90	2,494,473
	flatten	average	29.62%	50%	19.90	8,785,929
		concatenate	29.25%	**52%**	20.31	8,785,929
		maximum	30.74%	48%	19.68	8,785,929
	max-pool	average	29.25%	37%	16.56	2,494,473
		concatenate	24.44%	43%	16.86	2,625,545
		maximum	27.77%	50%	19.44	2,494,473
VGG16 + VGG19	avg-pool	average	31.11%	15%	11.52	4,853,257
		concatenate	31.48%	22%	13.37	4,984,329
		maximum	32.22%	18%	12.55	4,853,257
	flatten	average	32.59%	51%	20.89	11,144,713
		concatenate	29.25%	48%	19.31	17,567,241
		maximum	33.33%	**52%**	**21.33**	11,144,713
	max-pool	average	31.11%	45%	19.02	4,853,257
		concatenate	25.92%	47%	18.23	4,984,329
		maximum	26.66%	43%	17.41	4,853,257

Table 5.1: Performance comparison in terms of top-1 accuracy and coverage on the test set of HRA using various object-centric feature extractors and *early* fusion strategies.'Weighted Sum' refers to the derived criterion for finding the overall optimum. Bold font highlights the dominant performance across the same metric.

late fusion scheme where each ConvNet performs image recognition on its own and for final classification, softmax scores are combined by a mechanism which averages the prediction outputs, as illustrated in Figure 5.4.

	Operation on Conv. Base	Top-1 acc.	Coverage	Weighted Sum
VGG16+ResNet50	avg-pool	32.12%	30%	15.53
VGG19+ResNet50		**32.45%**	29%	15.36
VGG16+VGG19		30.88%	26%	14.22
VGG16+ResNet50	flatten	27.88%	41%	17.22
VGG19+ResNet50		26.14%	39%	16.28
VGG16+VGG19		27.88%	27%	13.72
VGG16+ResNet50	max-pool	29.13%	42%	**17.78**
VGG19+ResNet50		27.89%	**43%**	17.72
VGG16+VGG19		27.98%	31%	14.74

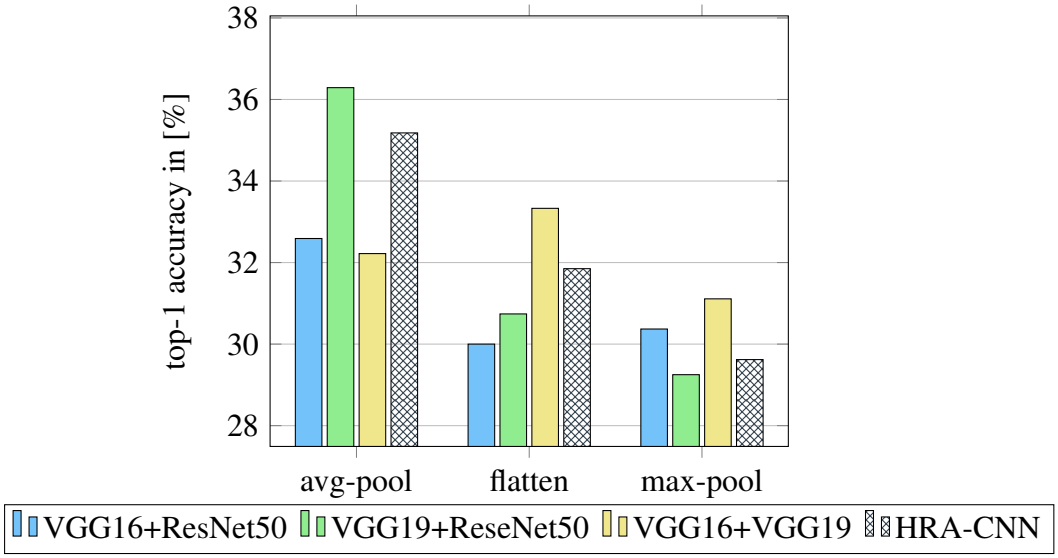
Table 5.2: Performance comparison in terms of top-1 accuracy and coverage on the test set of HRA using various object-centric feature extractors and *late* fusion strategies. 'Weighted Sum' refers to the derived criterion for finding the overall optimum. Bold font highlights the dominant performance across the same metric.

5.3.1 Results and Discussion

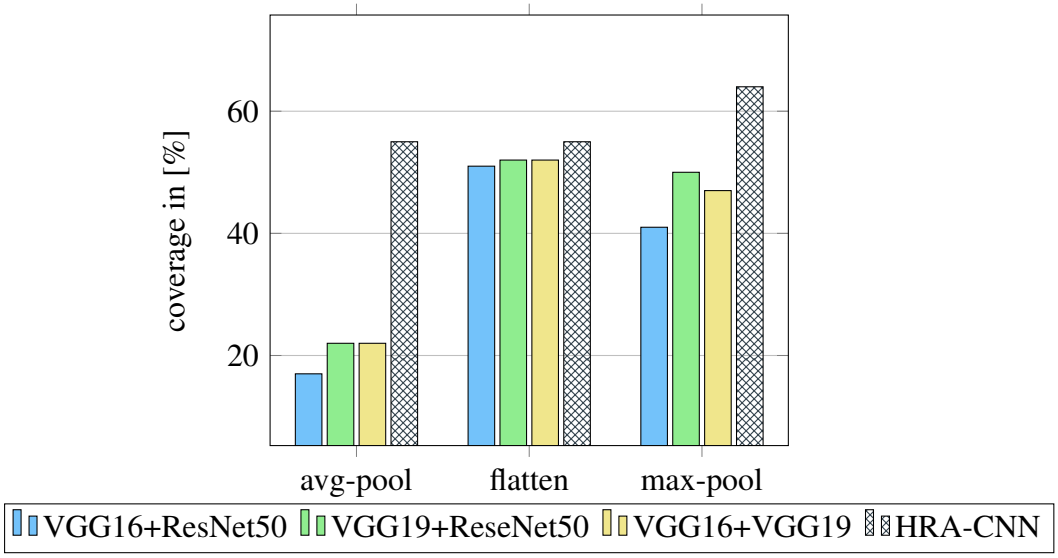
We compare results of three fusion and pooling operations and their combinations regarding object-centric features below.

Early fusion. Integrating different sources of information before they are processed by the target classier performs well in terms of top-1 accuracy, when compared to our two-phase transfer learning scheme[1] proposed in Chapter 4. The highest performing combination with respect to top-1 accuracy is *VGG19 + ResNet50*–utilising average pooling with concatenation strategy–which yields 36.29% as shown in Table 5.1, an absolute gain of 1.11% compared to the overall best performing HRA-CNN over the same pooling method reported in Table 4.3. This is mostly attributed to the fact that the combination of *VGG19 + ResNet50* has 2,494,473 trainable parameters, almost half of VGG16 fine-tuned model which has 4,853,257 trainable parameters. Similarly, we observe that the best performing combinations of the other two pooling methods, also outperform their individual counterparts as illustrated in Figure 5.5a. Interestingly, this is not the case when early fusion schemes are compared to their individual counterparts with respect to *coverage* performance metric introduced in section 4.3.3. This is a clear indication that although combination of object-centric features can improve top-1 accuracy, individual models consistently produce more robust predictions due to their simplistic design. The best performing combination of *VGG19 + ResNet50* reaches 52% coverage, which is an absolute drop of 12% from the best performing VGG16 reported previously, as seen in Figure 5.5b. Detailed results are reported in Table 5.1.

[1] denoted as HRA-CNN



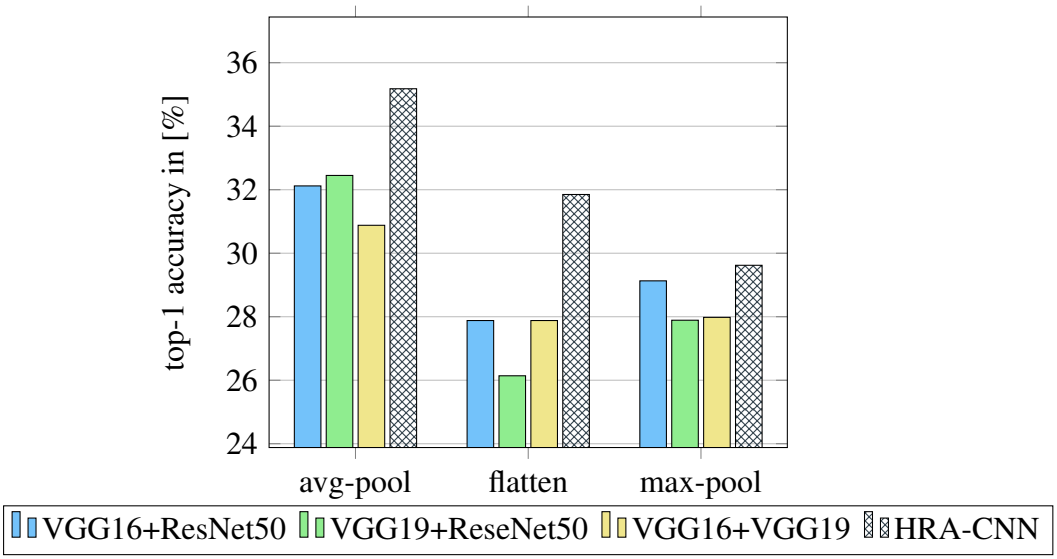
(a) Top-1 accuracy on the test set of HRA using our *early* fusion scheme of object-centric features.



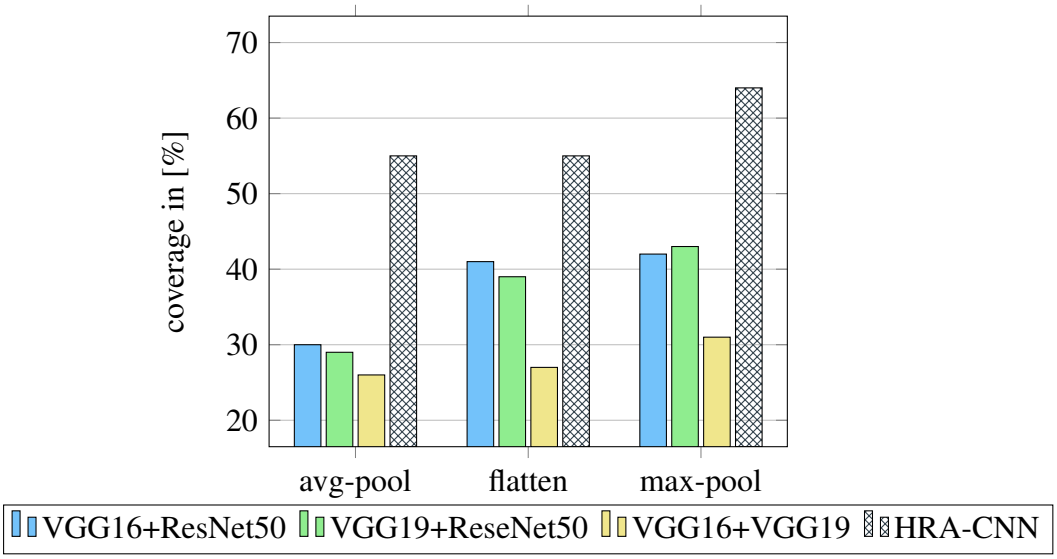
(b) Coverage on the test set of HRA using our *early* fusion scheme of object-centric features.

Figure 5.5: Comparative results in terms of top-1 accuracy and coverage for the three operations applied on the frozen convolutional base in our early fusion scheme of object-centric features. Only the best performing fusion strategy between *average*, *concatenation*, and *maximum* is illustrated alongside the best performing HRA-CNN for every operation applied on the frozen convolutional base during fine-tuning. Also, the best performing HRA-CNN for each pooling method is shown.

Late fusion. In contrast to early fusion, late fusion of object-centric features constantly trail their individual counterparts in most of the evaluations for both performance metrics as reported



(a) Top-1 accuracy on the test set of HRA using our *late* fusion scheme of object-centric features.



(b) Coverage on the test set of HRA using our *late* fusion scheme of object-centric features.

Figure 5.6: Comparative results in terms of top-1 accuracy and coverage for the three operations applied on the frozen convolutional base in our late fusion scheme of object-centric features.

in Table 5.2. The highest performing combination with respect to top-1 accuracy is *VGG19 + ResNet50*–utilising average pooling–which yields 32.45%, an absolute drop of 2.73% compared to the overall best performing HRA-CNN over the same pooling method reported in Table 4.3. Figure 5.6a suggests that for the best performing combinations of the other two pooling methods, HRA-CNN surpasses the combined models in the same way. This is due to the fact that concept learning precedes feature fusion in our late fusion scheme. Consequently,

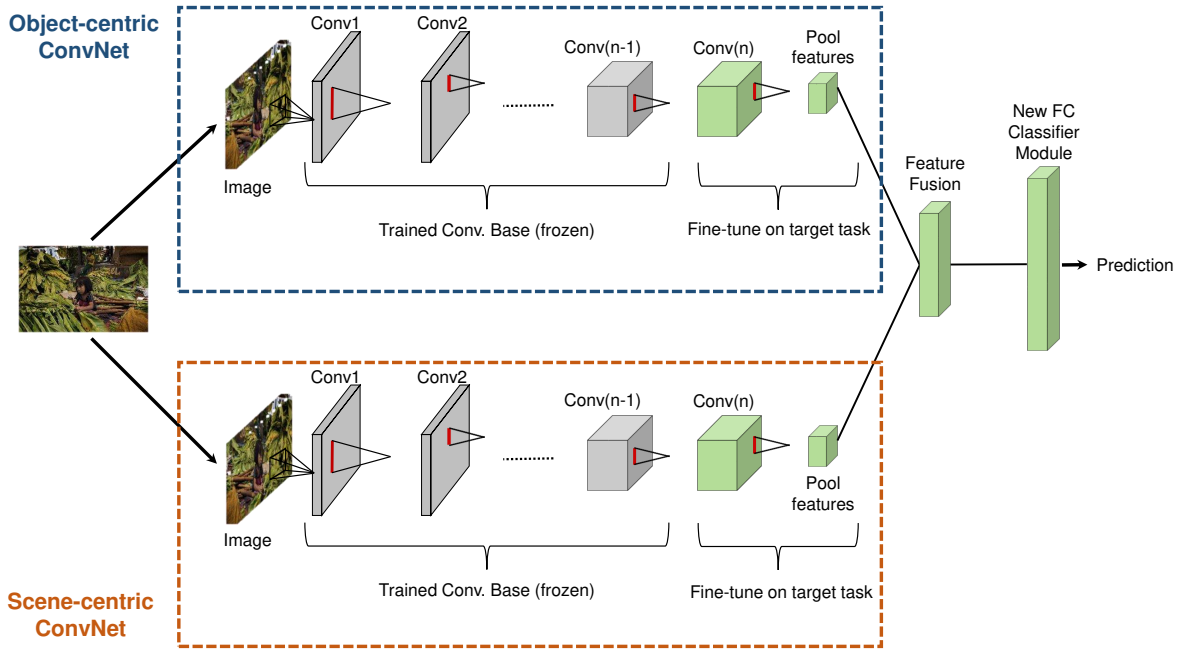


Figure 5.7: Illustration of our proposed object-centric and scene-centric high-level CNN *early* feature fusion and image classification system. The model consist of an object-centric feature extraction module (highlighted with blue border), a scene-centric feature extraction module (highlighted with orange border), a fusion network, and a classifier module for making predictions. Grey colour indicates the frozen convolutional layers, while the light green colour indicates the layers that will be randomly initialised.

classifiers are provided with information from diverse modalities, or features in a different way. For this reason correlations between those modalities might not be reflected in the classifier output scores. The same pattern of reduced performance compared to individual counterparts is observed for coverage. The best performing combination of *VGG19* + *ResNet50* reaches 43% coverage, which is a significant drop of 21% from the best performing VGG16 reported previously, as seen in Figure 5.6b. Our late fusion scheme appears to have some evident weaknesses compared to our early fusion schemes in the case of object-centric features.

5.4 Fusion of Object-Centric and Scene-Centric Deep Features

We showed that fusion of object-centric CNN features appears to have a negative effect on the performance of our HRA-CNN models, particularly in relation to coverage performance metric. This can be ascribed to the failure of ImageNet trained CNNs to effectively learn the

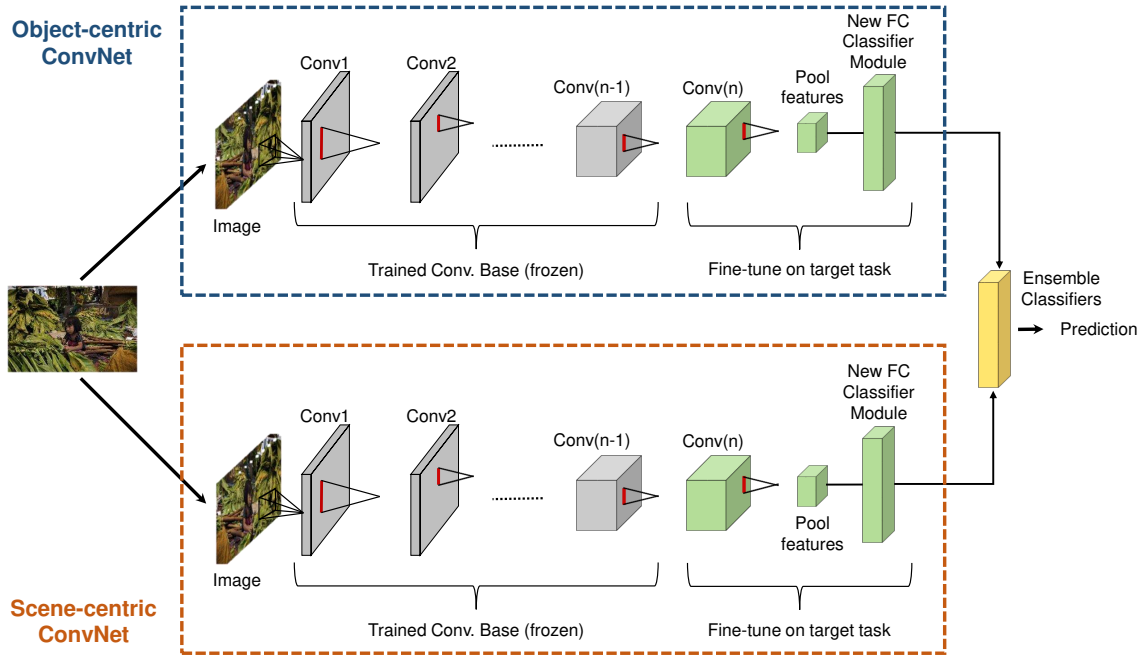


Figure 5.8: Illustration of our proposed object-centric and scene-centric high-level CNN *late* feature fusion and image classification system. The model consist of an object-centric feature extraction module (highlighted with blue border), a scene-centric feature extraction module (highlighted with orange border), and a module to ensemble classifiers before making predictions. Grey colour indicates the frozen convolutional layers, light green colour indicates the layers that will be randomly initialised, and light yellow colour indicates the ensemble of classifiers used to make the final prediction.

correspondence between these features or capture complementary cues. Recognition of human rights violations requires knowledge about both scenes and objects. As identified by visualising the class-discriminative regions of object-centric and scene-centric CNNs (see Figure 4.8) each ConvNet focuses on different aspects of the image in order to classify it. Based on that, we now investigate fusion of object-centric and scene-centric features for recognising potential human rights violations. We follow the same approach described in the previous section. However, this time instead of combining two different object-centric CNNs, for each test we combine an object-centric CNN with the scene-centric VGG16-Places365 [119]. Our proposed object-centric and scene-centric feature fusion and image classification systems are depicted in Figure 5.7 and Figure 5.8 for early and late fusion respectively.

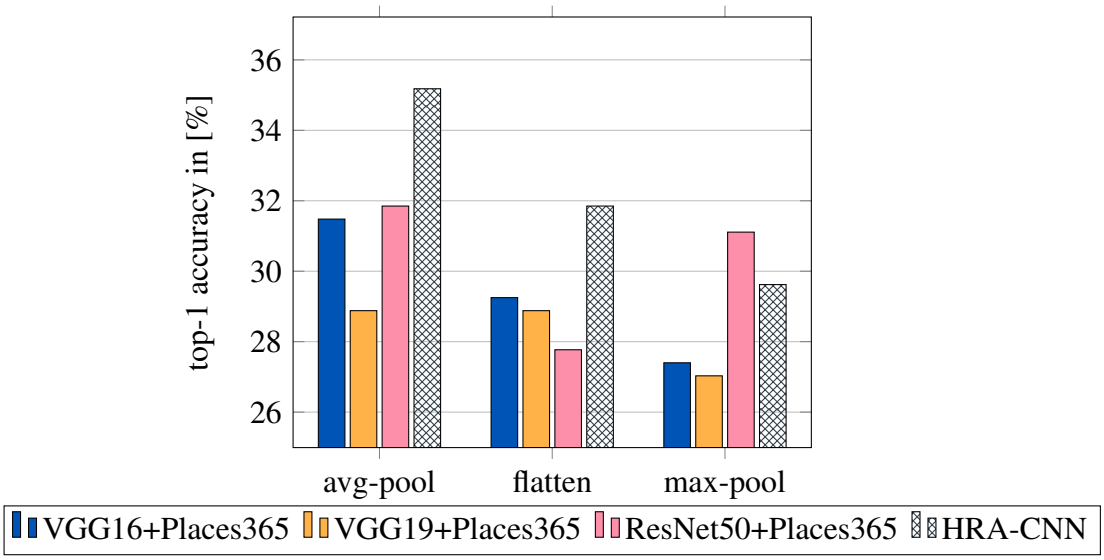
5.4.1 Differences and Complementarities

Early fusion. Remarkably, results indicate that early fusion of object-centric and scene-centric features trail their individual counterparts in most of the experiments both for top-1 accuracy

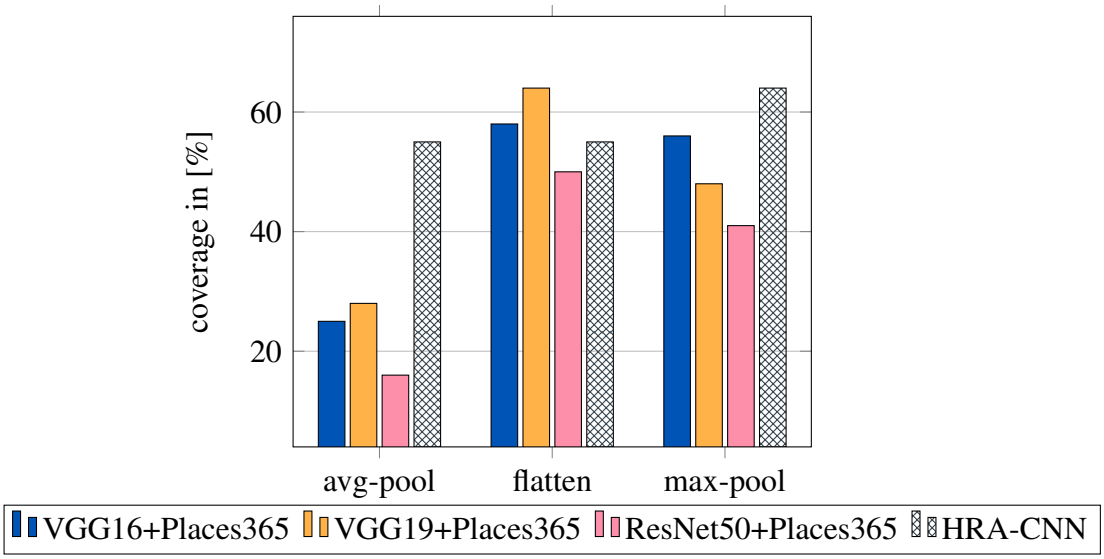
	Operation on Conv. Base	Fusion Strategy	Top-1 acc.	Coverage	Weighted Sum	Train Params.
VGG16 + Places365	avg-pool	average	31.48%	14%	11.37	4,853,257
		concatenate	30.37%	25%	13.84	4,984,329
		maximum	30.74%	19%	12.43	4,853,257
	flatten	average	27.40%	57%	21.1	11,144,713
		concatenate	27.77%	58%	21.44	17,567,241
		maximum	29.25%	54%	20.81	11,144,713
	max-pool	average	25.18%	45%	17.54	4,853,257
		concatenate	27.40%	49%	19.1	4,984,329
		maximum	24.44%	56%	20.11	4,853,257
VGG19 + Places365	avg-pool	average	27.03%	14%	10.25	4,853,257
		concatenate	27.77%	25%	13.19	4,984,329
		maximum	28.88%	28%	14.22	4,853,257
	flatten	average	25.55%	**64%**	**22.38**	11,144,713
		concatenate	28.88%	50%	19.72	17,567,241
		maximum	28.14%	51%	19.78	11,144,713
	max-pool	average	27.03%	37%	16	4,853,257
		concatenate	26.29%	47%	18.32	4,984,329
		maximum	26.29%	48%	18.57	4,853,257
ResNet50 + Places365	avg-pool	average	27.03%	5%	8	2,494,473
		concatenate	28.51%	14%	10.62	2,625,545
		maximum	**31.85%**	16%	11.96	2,494,473
	flatten	average	27.77%	41%	17.19	8,785,929
		concatenate	27.03%	50%	19.25	15,208,457
		maximum	24.81%	50%	18.70	8,785,929
	max-pool	average	25.92%	34%	14.98	2,494,473
		concatenate	25.55%	41%	16.63	2,625,545
		maximum	31.11%	30%	15.27	2,494,473

Table 5.3: Performance comparison in terms of top-1 accuracy and coverage on the test set of HRA using using various object-centric and scene-centric feature extractors and *early* fusion strategies. 'Weighted Sum' refers to the derived criterion for finding the overall optimum. Bold font highlights the dominant performance across the same metric.

and coverage, as illustrated in Figure 5.9. The highest performing combination with respect to top-1 accuracy is *ResNet50+Places365*–utilising average pooling and maximum fusion strategy–which yields 31.85%, a significant drop of 3.33% compared to the overall best performing HRA-CNN over the same pooling method reported in Table 4.3. Also the overall optimum of object-centric and scene-centric early fusion is 22.38, an absolute drop of 0.65 when compared to the overall optimum of HRA-CNNs. This came as a surprise, particularly if we consider the positive impact early fusion mechanism had in the case of object-centric-only features



(a) Top-1 accuracy on the test set of HRA using our *early* fusion scheme of object-centric and scene-centric features.



(b) Coverage on the test set of HRA using our *early* fusion scheme of object-centric and scene-centric features.

Figure 5.9: Comparative results in terms of top-1 accuracy and coverage for the three operations applied on the frozen convolutional base in our *early* fusion scheme of object-centric and scene-centric features. Only the best performing fusion strategy between *average*, *concatenation*, and *maximum* is illustrated for every pooling method. Also, the best performing HRA-CNN for each pooling method is shown.

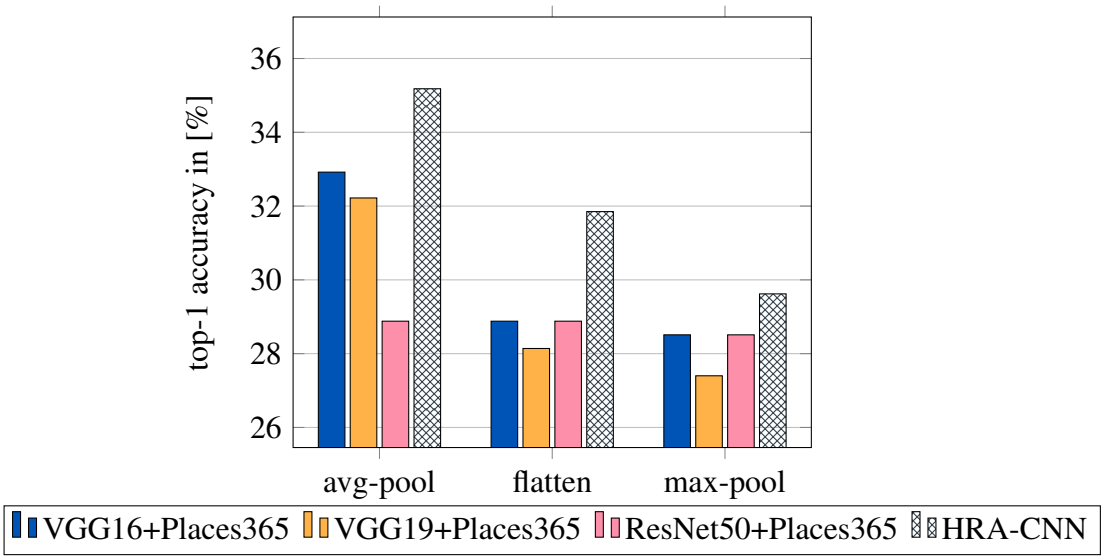
as depicted in Figure 5.5a. This is a clear indication that object-centric and scene-centric features, despite the fact that they focus on different visual cues, do not seem to complement

	Operation on Conv. Base	Top-1 acc.	Coverage	Weighted Sum
VGG16+Places365	avg	**32.92%**	31%	15.98
VGG19+Places365		32.22%	29%	15.30
ResNet50+Places365		28.88%	25%	13.47
VGG16+Places365	flatten	28.88%	42%	17.72
VGG19+Places365		28.14%	38%	16.53
ResNet50+Places365		28.88%	26%	13.72
VGG16+Places365	max	28.51%	41%	17.37
VGG19+Places365		27.40%	**44%**	**17.85**
ResNet50+Places365		28.51%	33%	15.37

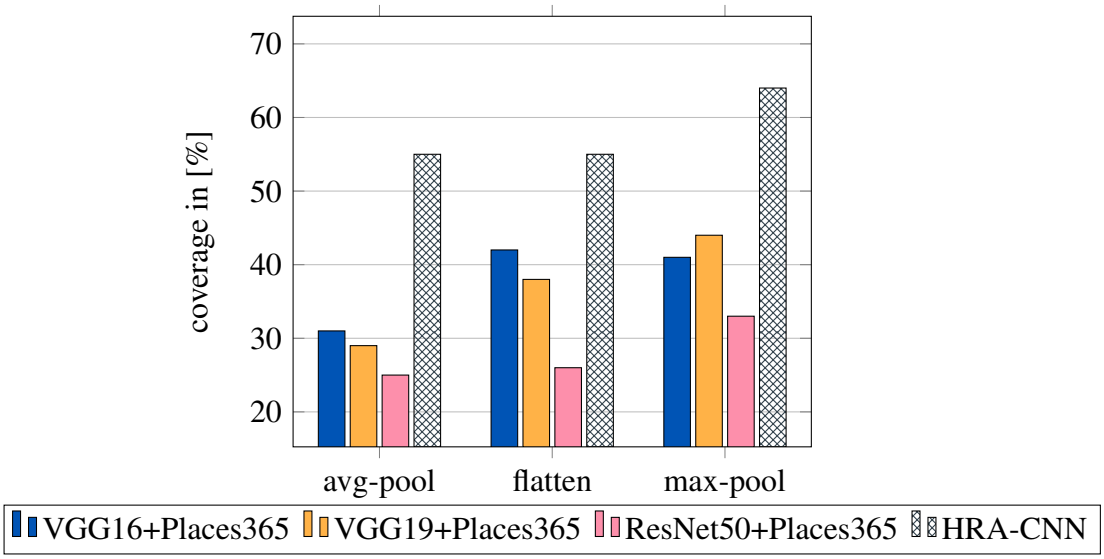
Table 5.4: Performance comparison in terms of top-1 accuracy and coverage on the test set of HRA using using various object-centric and scene-centric feature extractors and *late* fusion (ensemble of classifiers) strategies. 'Weighted Sum' refers to the derived criterion for finding the overall optimum. Bold font highlights the dominant performance across the same metric.

each other nearly as well as their individual counterparts. This can be attributed to the fact that merging features from object-centric and scene-centric CNNs results in more trainable parameters compared to object-centric fusion. Detailed results are reported in Table 5.3. One more interesting observation is that negative effects seem to occur mostly when combining very similar models like VGG16, VGG19 and VGG16-places365 which are all based on the same architecture. The best performing combination of *VGG19 + Places365* reaches 64% coverage, which is on par with the best performing HRA-CNN, while the rest of the best performing combinations trail their counterparts as seen in Figure 5.9b. Through the visualisation of the class-discriminative regions in Figure 5.11, we can have a better understanding of what has been learned inside the CNNs for the early fusion scheme.

Late fusion. Similar to object-centric features, late fusion of object-centric and scene-centric features constantly trail their individual counterparts in most of the evaluations for both performance metrics as reported in Table 5.4. Even the highest performing combinations report an absolute drop of 2.26% with respect to top-1 accuracy, illustrated in in Figure 5.10a. The same pattern of reduced performance compared to individual counterparts is observed for coverage. The best performing combination of *VGG19 + Places365* reaches 44% coverage, which is a significant drop of 20% from the best performing VGG16 reported previously, as seen in Figure 5.10b. The late fusion scheme appears to have weaknesses analogous to early fusion schemes of object-centric features.



(a) Top-1 accuracy on the test set of HRA using our *late* fusion scheme of object-centric and scene-centric features.



(b) Coverage on the test set of HRA using the *late* fusion scheme of object-centric and scene-centric features.

Figure 5.10: Comparative results in terms of top-1 accuracy and coverage for the three operations applied on the frozen convolutional base in our *late* fusion scheme of object-centric and scene-centric features.

5.4.2 Web-demo for Human Rights Violation Recognition

Based on our trained HRA-CNNs, we created a web-demo for HRVR accessible through computer or mobile device browsers. It is possible to upload photos to the web-based software

(a) object-centric **(b)** scene-centric **(c)** fusion

Figure 5.11: Informative regions for predicting the category `child labour` for CNNs pre-trained on different datasets using early fusion. Given an input image, we visualise the class-discriminative regions using Grad-CAM [85] for the output class. The object-centric models focus on the tools used by the young boys, the scene-centric models focus mostly on the head of the young boys, while the early fusion of the two CNNs focuses more on what the boys are holding (interaction with objects).

to identify if images depict a human right violation, while the system suggests the 3 most likely semantic categories from the HRA dataset. A screenshot of the prediction result on a web browser is shown in Figure 5.12. The Keras [8] python deep learning framework over TensorFlow [1] was used to train the back-end prediction model in the demo. With this system, those combating abuse will be able to go through images very quickly to narrow down the field and identify pictures which need to be looked at in more detail. Furthermore, with the extensive use of this software, we aim to collect an expanded range of images depicting human rights violations, in order to enhance the accuracy of our CNN models with larger data sets. Future directions for this work will include the capacity to receive feedback from people regarding the result.

5.5 Summary and Limitations

In this chapter we have presented a thorough investigation on the relevance of features extracted from CNNs trained on different image datasets, for predicting potential human rights violations. Based on two main fusion strategies, *early* and *late*, we have proposed various, complete visual human rights violation recognition systems. First, we combine CNN features from fine-tuned

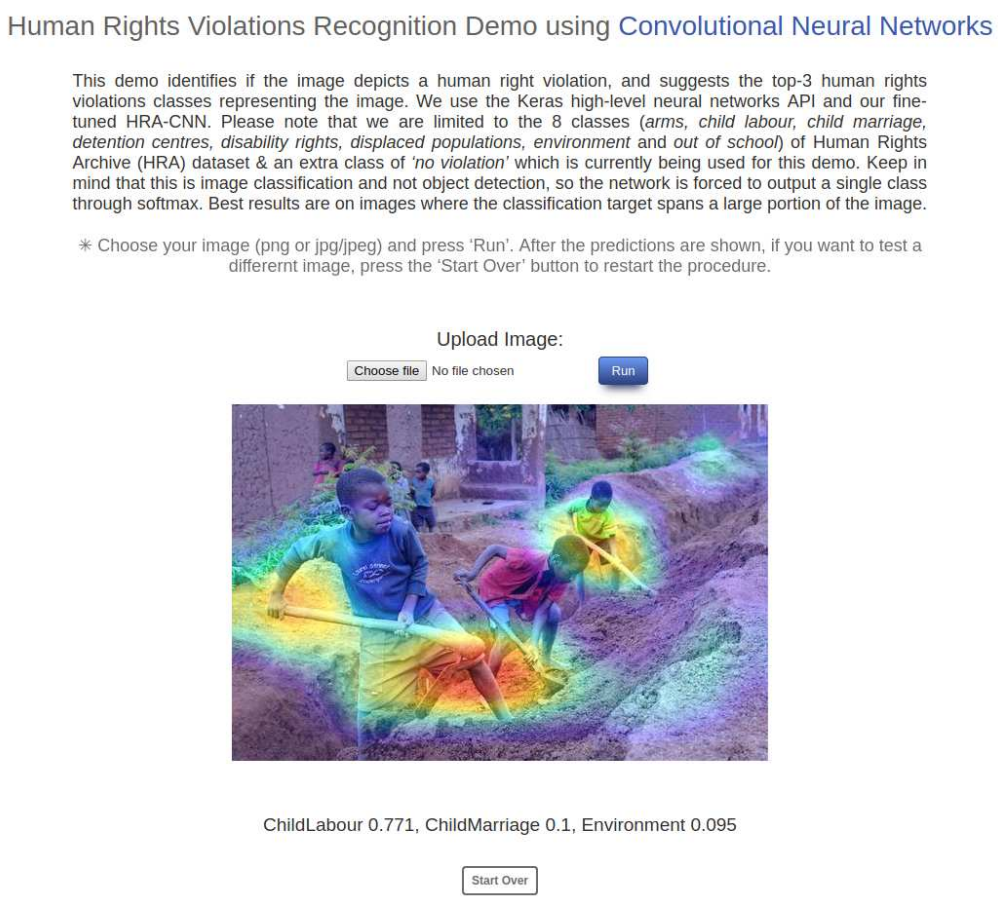


Figure 5.12: A screenshot of the human rights violation recognition demo based on our fine-tuned HRA-CNN. The web-demo predicts the type of human right that is being violated for uploaded photos.

models on object-centric databases. Early fusion in this case operates well for top-1 accuracy, achieving higher performance in some instances compared to their individual counterparts. However, this is not the case for the coverage metric, where combined features constantly trail the individual models by a large margin. This shows that for the HRVR task robust predictions are essential, while it is possible to trade coverage for accuracy by refusing to process some examples. Performance gap between individual models and fusion schemes increases even further when late fusion is tested. Following this observation, we turned our attention to combining features from models pretrained on different datasets, ImageNet (object-centric) and Places (scene-centric), in order to learn those complementary cues required for the task of visual recognition of human rights violations. Remarkably, experimental results revealed that even refined combination of features fails to outperform the simplistic design of their individual counterparts for both fusion strategies. These results reinforce the view that although prediction

of human rights violations consists of underlying tasks such as object and scene recognition, it poses a challenge at a higher level for the representation learning methods. Observations made in this chapter clearly reflect that recognising individual objects or scenes is just a first step for machines to comprehend human rights violations in the visual world.

Chapter 6

Recognising Displaced People from Images by Exploiting their Dominance Level

The displacement of people refers to the forced movement of people from their locality or environment and occupational activities, and it is seen as a form of social change due to a number of factors such as *armed conflict*, *violence*, *persecution*, and *human rights violations*. Globally, there are now almost 68.5 million forcibly displaced people–roughly equivalent to the entire UK population–while today 1 out of every 110 people in the world is displaced. Despite those figures, human rights analysts and advocates still rely on manual labour to analyse human rights-related imagery and then act accordingly. Computer vision can help automate parts of this process and turn recognition of displaced populations into a potent service that could improve humanitarian responses. Our intuition is that a person's control level of a situation can be a notifying difference between the encoded visual content of an image that depicts a non-violent situation and the encoded visual content of an image displaying displaced people. In this chapter, after introducing a score that can describe an entire image based on all individuals' control level of the situation, typical image classification is extended within a novel, uniform framework which infers potential displaced people from images.

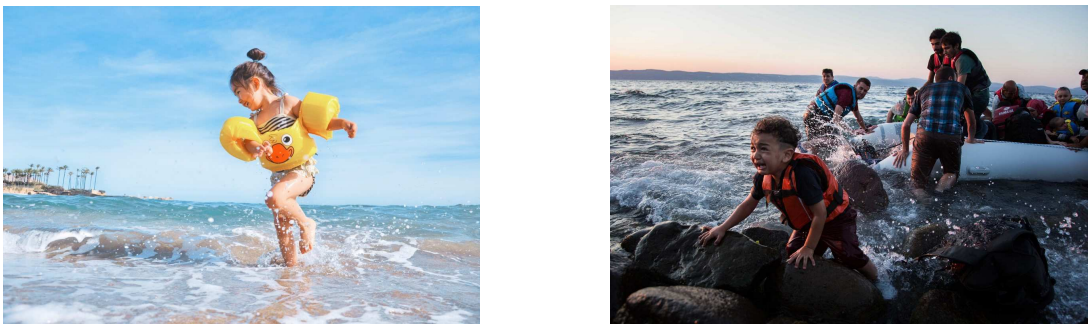
6.1 Introduction

The previous two chapters were focused on the multiclass classification problem of human rights violations. In this chapter we change direction and study one of the most reported modern violations against human rights, *displaced people*. We will explore the task of labelling still images as either `displaced people` or `non-displaced people`. This is challenging as classification schemes based on object detection or scene recognition regularly fail to discriminate the encoded visual content of an image that depicts a non-violent situation and the encoded visual content of an image displaying displaced people, as discussed in previous chapters. We make the following contributions: (i) depending on the deep visual representations used we show that performance can be improved by exploiting dominance level of people. (ii) we introduce a new method for interpreting the overall dominance level of an entire image sample based on the emotional states of all individuals on the scene; (iii) we introduce a dataset for learning and evaluating this problem; We compare our method with fine-tuned CNN representations for this task.

6.2 Motivation and Approach

The motivation of this work comes from the official figures concerning forcibly displaced people published by the Office of the United Nations High Commissioner for Refugees (UNHCR) in their statistical yearbooks. The displacement of people refers to the forced movement of people from their locality or environment and occupational activities[1]. It is a form of social change caused by a number of factors such as armed conflict, violence, persecution and human rights violations. Every year millions of men, women and children are forced to leave their homes and seek refuge from wars, human rights violations, persecution, and natural disasters. The number of forcibly displaced people came at a record rate of 44,400 every day throughout 2017, raising the cumulative total to 68.5 million at the year's end, overtaken the total population of the United Kingdom [100]. Up to 85% of the forcibly displaced people find refuge in low- and middle-income countries, calling for increased humanitarian assistance worldwide. To reduce the amount of manual labour required for human-rights-related image analysis, we introduce `DisplaceNet`, a novel model which infers potential displaced people from images by integrating the control level of the situation and conventional CNN classifier into one framework for binary image classification.

[1] A distinction is often made between conflict-induced and disaster-induced displacement, yet the lines between them may be blurred in practice.



(a) Child playing **(b)** Displaced people

Figure 6.1: Inferring potential displaced people only from object detection and/or scene recognition is condemned to failure. Displaced people recognition poses a challenge at a higher level for the well-studied, deep image representation learning methods. Regularly, emotional states can be a notifying difference between the encoded visual content of an image that depicts a non-violent situation and the encoded visual content of an image displaying displaced people.

In the era of social media and big data, the use of visual evidence to document conflict and human rights abuse has become an important element for human rights organisations and advocates. However, the omnipresence of visual evidence may deluge those accountable for analysing it. Currently, information extraction from human-rights-related imagery requires manual labour by human rights analysts and advocates. Such analysis is time consuming, expensive, and remains emotionally traumatic for analysts to focus on images of horrific events. In this work, we strive to reconcile this gap by automating parts of this process; given a single image we label the image as either *displaced people* or *non displaced people*. Figure 6.1 illustrates that naive schemes based solely on object detection or scene recognition are doomed to fail in this binary classification problem. If we can exploit existing smartphone cameras, which are ubiquitous, it may be possible to turn recognition of displaced populations into a powerful and cost-effective computer vision application that could improve humanitarian responses.

6.3 Implementation Details

Two-stage fine-tuning of deep CNNs has shown the potential to address the multi-class classification problem of HRVR, but only to a certain extent as shown in Chapter 4. In this chapter, we introduce `DisplaceNet`, a novel method designed with a human-centric approach for solving a sought-after, binary classification problem in the context of human rights image analysis; *displaced people recognition*. As reported by Kosti *et al.* [51], distribution of dominance values across emotion categories of their EMOTIC dataset shows that people are not in control when

they show emotion categories like *Suffering*, *Pain*, *Sadness* whereas when the Dominance is high, emotion categories like *Esteem*, *Excitement*, *Confidence* occur more often. Additionally, places where people usually show high Dominance are sport-related places and sport-related attributes. On the contrary, low Dominance is shown in Jail Cell or attributes like Enclosed Area or Working, where the freedom of movement is restricted. This resonates well with our own common sense in judging potential displacement cases. Accordingly, our hypothesis is that the control level of the situation by the person, ranging from *submissive* / *non-control* to *dominant* / *in-control*, is a powerful cue that can help our network make a distinction between displaced people and non-violent instances.

First, we develop an end-to-end model for recognising rich information about people's emotional states by jointly analysing the person and the whole scene. We use the continuous dimensions of the *VAD Emotional State Model* [69], which describe emotions using three numerical dimensions: Valence (V); Arousal (A); and Dominance (D). In the context of this work, we have focused only on dominance–measures the control level of the situation by the person–because it is considered as the most relevant for the task of recognising displaced people. Second, following the estimation of emotional states, we introduce a new method for interpreting the overall dominance level of an entire image sample based on the emotional states of all individuals on the scene. As a final step, we propose to assign weights to image samples according to the image-to-overall-dominance relevance to guide prediction of the image classifier.

6.4 Method

Our method enables recognition of displaced people by exploiting the dominance level of an entire image. Our goal is to label challenging everyday photos as either 'displaced people' or 'non displaced people'.

First, in order to detect the emotional traits of a given image, we need to accurately localise the box containing a *human* and the associated object of interaction (denoted by b_h and b_o, respectively), as well as identify the emotional states e of each human using the VAD model. Our proposed solution adopts the RetinaNet [60] object detection framework alongside an additional *human-centric* branch that estimates the continuous dimensions of each detected person and then determines the overall dominance level of the given image.

Specifically, given a set of candidate boxes, RetinaNet outputs a set of object boxes and a class label for each box. While the object detector can predict multiple class labels, our model is concerned only with the 'person' class. The region of the image comprising the person whose feelings are to be estimated at b_h is used alongside the entire image for simultaneously

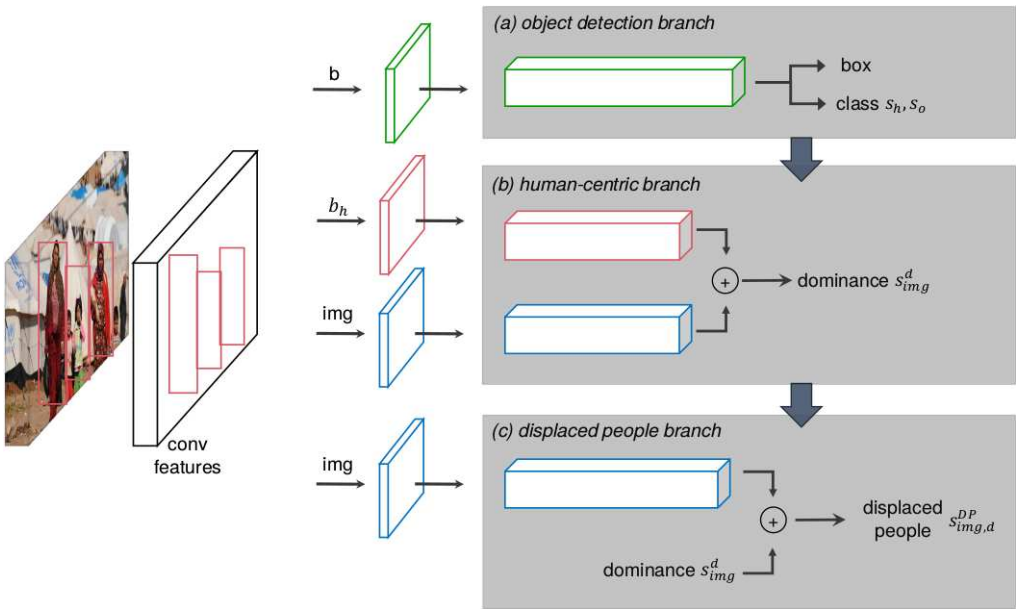


Figure 6.2: DisplaceNet architecture. Our model consists of (a) an *object detection* branch, (b) *human-centric* branch, and (c) a *displaced people* branch. The image features and their layers are shared between the human-centric and displaced people branches (blue boxes).

extracting their most relevant features. These features, are fused and used to perform continuous emotion recognition in VAD space. Our model extends typical image classification by assigning a triplet score $s^{DP}_{img,d}$ to pairs of candidate human boxes b_h and the displaced people category a. To do so, we decompose the triplet score into three terms:

$$s^{DP}_{img,d} = s_h \cdot s^d_{h,img} \cdot s^{DP}_{img} \tag{6.1}$$

We discuss each component next, followed by details for training and inference. The overall architecture of DisplaceNet is shown in Figure 6.2.

6.4.1 Model components

Object detection branch. The object detection branch of DisplaceNet is identical to that of RetinaNet [60] single stage classifier. First, an image is forwarded through ResNet-50 [31], then in the subsequent pyramid layers, the more semantically important features are extracted and concatenated with the original features for improved bounding box regression.

Human-centric branch. The first role of the human-centric branch is to assign an emotion classification score to each human bounding box. Similar to [52], we use an end-to-end model with three main modules: two feature extractors and a fusion module. The first module takes the region of the image comprising the person whose emotional traits are to be estimated, b_h, while

the second module takes as input the entire image and extracts global features. This way the required contextual support is accommodated in the emotion recognition process. Finally, the third module takes as input the extracted image and body features and estimates the continuous dimensions in VAD space.

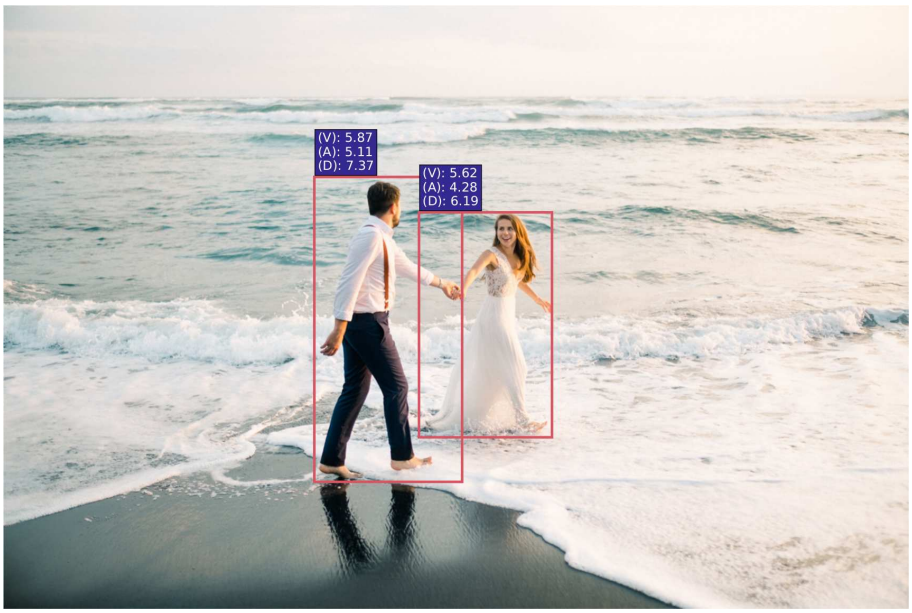
The second role of the human-centric branch is to assign a dominance score s^d_{img} which characterises the entire input image. s^d_{img} is the encoding of the overall dominance score relative to human box b_h and entire image *img*, that is:

$$s^d_{img} = \frac{1}{n}\sum_{i=1}^{n} s^d_{h,img} \tag{6.2}$$

Figure 6.3a and Figure 6.4a illustrate the three different emotional states over the estimated target objects locations while Figure 6.3b and Figure 6.4b shows the overall dominance score proposed here. Note that although all three predicted numerical dimensions are depicted, only dominance is considered to be the most relevant to the task of recognising displaced people.

Displaced people branch. The first role of the displaced people branch is to assign a classification score to the input image. Similar to two-phase transfer learning scheme introduced in 4.3, we train an end-to-end model for binary classification ('displaced people' or 'non displaced people') of everyday photos. In order to improve the discriminative power of our model, the second role of the displaced people branch is to integrate s^d_{img} in the recognition pipeline. Specifically, the raw image classification score is readjusted based on the inferred dominance score. Each dominance unit, that is deltas from the neutral state, is expressed as a numeric weight varying between 1 and 10, while the neutral states of dominance are assigned between 4.5 and 5.5 based on the number of examples per each of the scores in the continuous dimensions reported in [52]. The adjustment that will be assigned to the raw probability, s^{DP}_{img} is the weight of dominance multiplied by a factor of 0.11 which has been experimentally set. When the input image depicts positive dominance, the adjustment factor is subtracted from the positive displaced people probability and added to the negative displaced people probability. Similarly, when the input image depicts negative dominance the adjustment factor is added to the negative displaced people probability and subtracted from the positive displaced people probability. This is formally written in Algorithm 1. Finally, in instances where no b_h are detected from the object detection branch, (6.1) is reduced into plain image classification as follows:

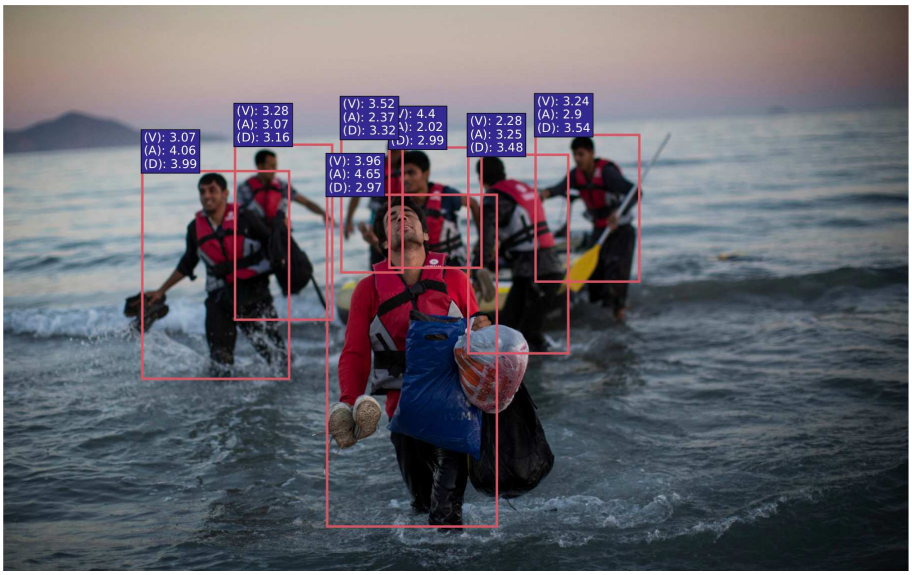
$$s^{DP}_{img,d} = s^{DP}_{img} \tag{6.3}$$



(a) Continuous emotion recognition in VAD space

(b) Our proposed *overall dominance score*

Figure 6.3: Example of estimating continuous emotions in VAD space vs the proposed overall dominance score that characterises an entire image based on all individuals' control level of the situation from the combined body and image features. (a) shows the predicted emotional states and their scores from the person region of interest (RoI), while (b) shows the same images analysed with our proposed *overall dominance score*. The dominance score will be later integrated with the standard image classification scores s_{img}^{DP} to identify displaced people.



(a) Continuous emotion recognition in VAD space

(b) Our proposed *overall dominance score*

Figure 6.4: Further example of estimating continuous emotions in VAD space vs the proposed overall dominance score that characterises an entire image based on all individuals' control level of the situation from the combined body and image features.

6.4.2 Training

Due to different datasets, convergence times and loss imbalance, all three branches have been trained separately. For object detection we adopted an existing implementation of the RetinaNet object detector, pre-trained on the COCO dataset [61], with a ResNet-50 backbone.

Algorithm 1: Calculate $s^{DP}_{img,d}$

Require: $b_h > 0$

$s_{pos} \leftarrow s^{dp}_{img}$ {dp: positive displaced people}
$s_{neg} \leftarrow s^{ndp}_{img}$ {ndp: negative displaced people}
if $weight \geq 4.5$ **and** $weight \leq 5.5$ **then**
  Return s_{pos}, s_{neg}
else if $weight > 5.5$ **then**
  $diff = weight - 5.5$
  $adj = diff * 0.11$
  $s_{pos} = s_{pos} - adj$
  $s_{neg} = s_{neg} + adj$
else if $weight < 4.5$ **then**
  $diff = 4.5 - weight$
  $adj = diff * 0.11$
  $s_{pos} = s_{pos} + adj$
  $s_{neg} = s_{neg} - adj$
end if
Return s_{pos}, s_{neg}

For emotion recognition in continuous dimensions, we formulate this task as a regression problem using the Euclidean loss (L2 loss). The Euclidean loss relies on the Euclidean distance between two vectors - the prediction and the ground truth. The distance is calculated by taking the square root of the sum of the squared pair-wise distances of every dimension, $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$. The two feature extraction modules are designed as truncated versions of various well-known CNNs and initialised using pretrained models on two large-scale image classification datasets, ImageNet [54] and Places [119]. The truncated version of those CNNs removes the fully connected layer and outputs features from the last convolutional layer in order to maintain the localisation of different parts of the images which is significant for the task at hand. Features extracted from these two modules (red and blue boxes in Figure 6.2b are then combined by a fusion module. This module first uses a global average pooling layer to reduce the number of features from each network and then a fully connected layer, with an output of a 256-D vector, functions as a dimensionality reduction layer for the concatenated pooled features. Finally, we include a second fully connected layer with 3 neurons representing valence, arousal and dominance. The pipeline of the model is shown in Figure 6.5. The parameters of the three modules are learned jointly using stochastic gradient descent with momentum of 0.9. The batch size is set to 54 and we use dropout with a ratio of 0.5.

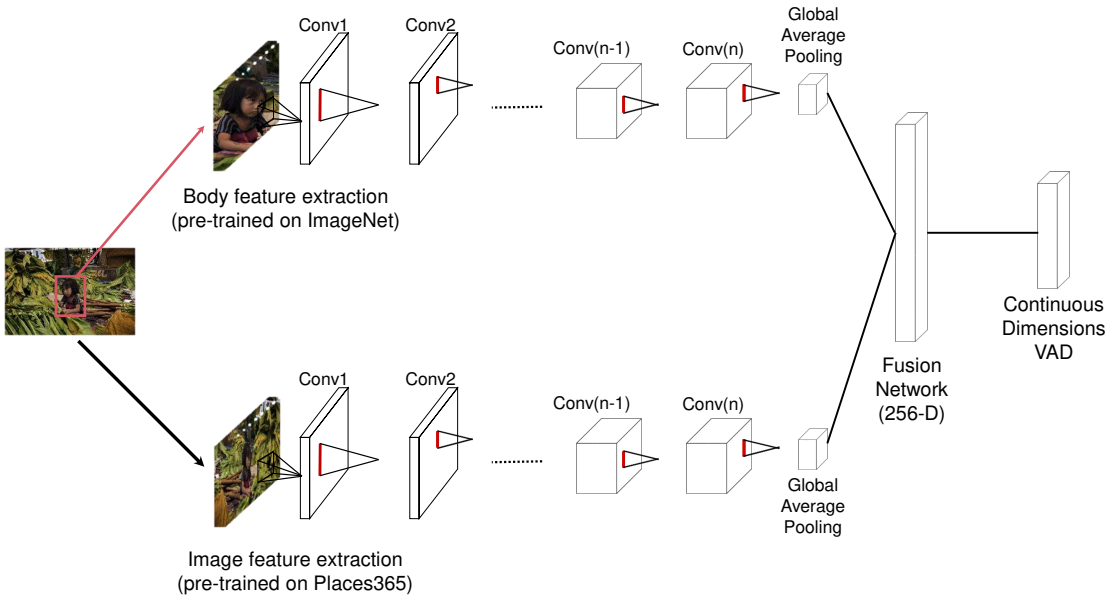


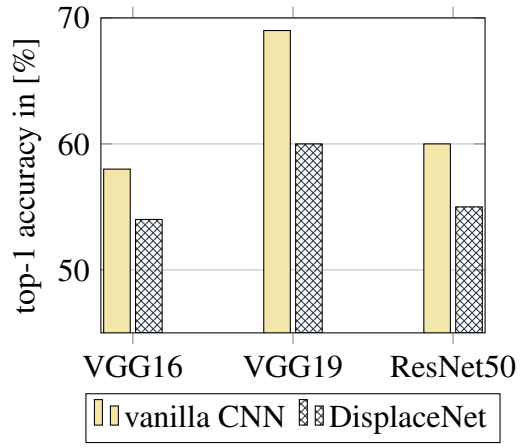
Figure 6.5: End-to-end model for emotion recognition in context. The model consists of two feature extraction modules and a fusion network for jointly estimating the discrete categories and the continuous dimensions, similar to [52].

We formulate displaced people recognition as a binary classification problem. We train an end-to-end model for classifying everyday images as displaced people-positive or displaced people-negative, based on the context of the images. We fine-tune various CNN models for this two-class classification task. First, we conduct feature extraction utilising only the convolutional base of the original networks in order to end up with more generic representations as well as retaining spatial information similar to emotion recognition pipeline. The second phase consists of unfreezing some of the top layers of the convolutional base and jointly training a newly added fully connected layer and these top layers. All the CNNs[2] presented here were trained using the Keras Python deep learning framework [8] over TensorFlow [1] on Nvidia GPU P100.

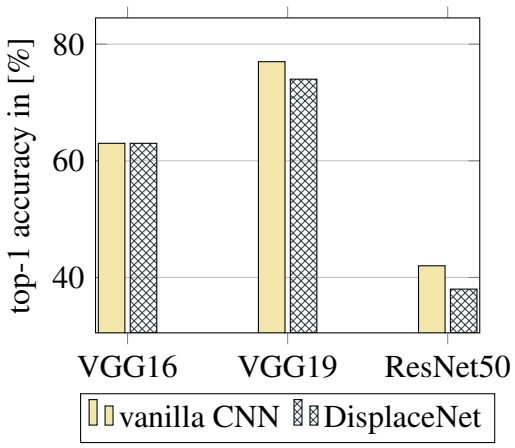
6.5 Experiments

Our emotion recognition implementation is based on the emotion recognition in context (EMOTIC) model [52], with the difference that our model estimates only continuous dimensions in VAD space. We train the three main modules on the EMOTIC database, which contains a total number of 18,316 images with 23,788 annotated people, using pre-trained CNN feature

[2]Available at https://github.com/GKalliatakis/DisplaceNet

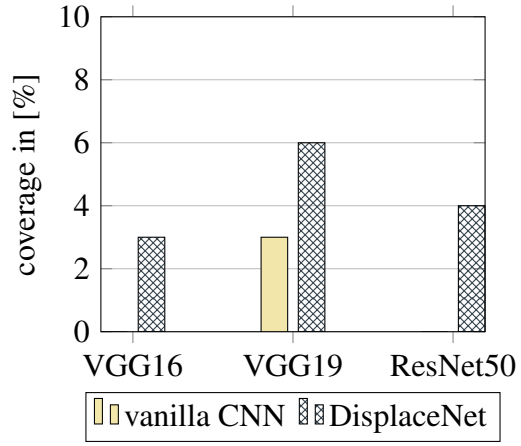


(a) Top-1 accuracy on the test set of HRA–Binary with one layer of the backbone network fine-tuned.

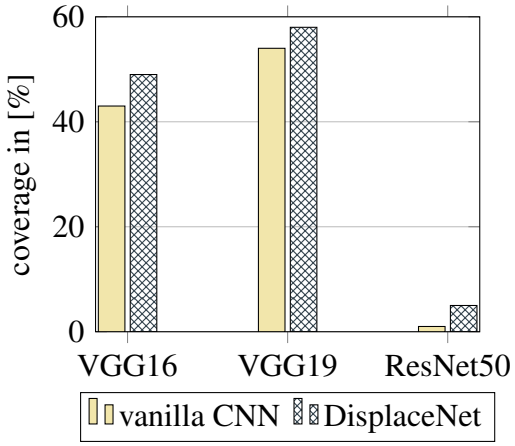


(b) Top-1 accuracy on the test set of HRA–Binary with two layers of the backbone network fine-tuned.

Figure 6.6: Comparative results in terms of top-1 accuracy for fine-tuned models (vanilla CNN) and our proposed method, *DisplaceNet*, over various backbone networks.



(a) Coverage on the test set of HRA–Binary with one layer of the backbone network fine-tuned.



(b) Coverage on the test set of HRA–Binary with two layers of the backbone network fine-tuned.

Figure 6.7: Comparative results in terms of coverage for fine-tuned models (vanilla CNN) and our proposed method, *DisplaceNet*, over various backbone networks. Note that in some instances vanilla CNNs achieve a coverage of 0%.

extraction modules. We treat this multiclass-multilabel problem as a regression problem by using a weighted Euclidean loss to compensate for the class imbalance of EMOTIC.

For the classification part, we fine-tune our models for 50 iterations on the HRA subset with a learning rate of 0.0001 using the stochastic gradient descent (SGD) [55] optimizer for cross-entropy minimization. These *vanilla* models will be examined against DisplaceNet. Here,

backbone CNN	layers fine-tuned	vanilla CNN			DisplaceNet		
		Top-1 acc.	Coverage	Weighted Sum	Top-1 acc.	Coverage	Weighted Sum
VGG16	1	58%	0%	14.5	54%	**3%**	14.25
VGG19		69%	3%	18	60%	**6%**	16.5
ResNet50		60%	0%	15	55%	**4%**	14.75
VGG16	2	63%	43%	26.5	63%	**49%**	28
VGG19		77%	54%	32.75	74%	**68%**	**35.5**
ResNet50		42%	1%	10.75	38%	**5%**	10.75
mean	-	61.5%	16.83%	19.58	57.33%	**20.83%**	19.54

Table 6.1: Detailed results on displaced people recognition using DisplaceNet. We show the main baseline and DisplaceNet for various network backbones. 'Weighted Sum' refers to the derived criterion for finding the overall optimum. We bold the leading entries on coverage.

vanilla means pure image classification using solely fine-tuning without any alteration. To enable a fair comparison between vanilla CNNs and DisplaceNet, we use the same backbone combinations for all modules described in Figure 6.2.

The main test platform on which we could demonstrate the effectiveness of DisplaceNet and analyse its various components is the `HRA-Binary Dataset` (Section 3.3.5). The *displaced population* category of that dataset contains 609 images of displaced people and the same number of non displaced people counterparts for training, as well as 50 images collected from the web for testing and validation, as described in Section 3.3.5. We evaluate DisplaceNet with the same two metrics utilised before, *top-1 accuracy* and *coverage*, and compare its performance against the sole use of a CNN classifier.

Quantitative Results

We report comparisons in both *top-1 accuracy* and *coverage* metrics for fine-tuning up to two convolutional layers in order to be consistent with the implementation introduced in Chapter 4.3. The per-network results are shown in Table 6.1. The implementation of vanilla CNNs is solid with mean accuracy of 61.5% accuracy, and mean coverage of 16.83%. Interestingly, some models yield a 0% coverage, which proves yet again that it is possible to trade coverage with accuracy in the context of human rights image analysis. In any case, vanilla CNNs provide a strong baseline to which we will compare our method.

DisplaceNet has a mean accuracy of 57.33% which is a minor drop of 4.17 points over the strong baselines of 61.5% achieved by vanilla CNNs. This indicates a relative loss of only 6.7%. Comparative results for all cases are illustrated in Figure 6.6. We believe that this negligible drop in accuracy is mainly due to the fact that the *test* set is not solely made up of images with

(a)

(b)

Figure 6.8: Examples of recognising displaced people with DisplaceNet. In this instance, DisplaceNet overturns the initial-false prediction of the vanilla CNN.

people in their context, it also contains images of generic objects and scenes, where only the sole classifier's prediction is taken into account. Concerning coverage performance metric, DisplaceNet achieves a mean percentage of 20.83%. This is an absolute gain of 4 points over

the baseline of 16.83%. This is a significant relative improvement of 23.76%. It is evident from Figure 6.7 that DisplaceNet constantly improves the coverage performance of the system, even for extreme cases where vanilla CNNs fail to produce robust predictions at all (VGG16 and ResNet 50 in Figure 6.7a). In order to reach this level of coverage, DisplaceNet sacrifices top-1 accuracy. Coverage thus became the main performance metric optimised during this task, with mean top-1 accuracy held at 57.33%.

Qualitative Results.

We show our displaced people recognition results in Figure 6.8 and Figure 6.9. Each image shows two predictions alongside their probabilities. Top prediction is given by DisplaceNet, while the bottom prediction is given by the respective vanilla CNN classifier. Green colour implies that no displaced people were detected, while red colour signifies that potentially displaced people were detected. Our method can successfully classify displaced people by overturning the initial-false prediction of the vanilla CNN (Figure 6.8). Moreover, DisplaceNet can strengthen the initial-true prediction of the sole classifier (Figure 6.9). Finally, our method can be incorrect, because of false dominance score inferences. Some of them are caused by a failure of continuous dimensions emotion recognition, which is an interesting open problem for future research.

6.6 Summary

In this chapter we have presented a human-centric approach for recognising displaced people from still images. This two-class labelling problem is not trivial, given the high-level image interpretation required. Understanding a person's control level of the situation from his frame of reference is closely related with situations where people have been forcibly displaced. Thus, the key to our computational framework is people's dominance level, which resonates well with our own common sense in judging potential displacement cases. We introduce the overall dominance score of the image which is responsible for weighting the classifier's prediction during inference. We benchmark performance of our DisplaceNet model against sole CNN classifiers. Our experimental results showed that this is an effective strategy, which we believe has good potential beyond human rights related classification. We hope this work will spark interest and subsequent work along this line of research.

(a)

(b)

Figure 6.9: Further examples of recognising displaced people with DisplaceNet. In this instance, DisplaceNet strengthens the initial-true prediction, resulting in higher coverage.

Chapter 7

Harnessing Global Emotional Traits for Two-Class Human Rights Abuse Classification

---

Modern deep learning systems can learn to detect many of the kinds of objects that are of interest to human rights researchers, including tanks, missiles, helicopters, aeroplanes, military vehicles, particular styles of building, and large crowds. They can also detect visually distinct geographic locations like bridges over water, mountainous terrain, or a desert. From a computer vision perspective, human rights violation recognition from a given image is computationally complex because it involves the detection of numerous semantic concepts (*i.e.*, objects, and scenes) taking place in a dynamic environment. The kinds of images relevant to human rights investigations are significantly more complex, making it hard to identify potential violations only by properties of the surrounding scene or the related objects. We have shown that people's emotional states are a contributing factor to the automated understanding of visual information of human rights violations in a binary classification context. Taking this into account, we introduce a robust image characterisation mechanism that will help us tackle the two most widespread human rights violations in the era of social media and big data, *child labour*, and *displaced populations*.

7.1 Introduction

In the previous chapter, we investigated binary image classification of real-world images for recognising displaced people using the dominance level of the identified people in a scene. In this chapter, we expand on those findings by establishing and addressing two different human rights-related, visual recognition scenarios, in challenging everyday photos: *child labour*, and *displaced populations*. We make the following contributions beyond those described in Chapter 6 : (i) we show that coverage performance can further be improved by exploiting two emotional states instead of a single one; (ii) we introduce a more robust mechanism capable of characterising an image based on two emotional states of all people in a scene; (iii) we present a human-centric, end-to-end model for recognising two types of human rights violations; (iv) we introduce a dataset for learning and evaluating those two different tasks. We compare our method both with the fine-tuned CNN representations (Chapter 4) and DisplaceNet (Chapter 6), for overlapping categories.

7.2 Motivation and Approach

The motivation of this work comes from the promising results of DisplaceNet reported in the previous chapter. Experiments with DisplaceNet revealed that people's dominance level resonates well with our common sense in evaluating certain situations as potentially displaced populations or non-displaced populations. Here, we want to build on those findings and further exploit emotional traits this time employing and combining two at the same time, before testing them in two different settings. This chapter attempts to address the problem of human rights abuse prediction from a single image, for two independent scenarios. Note that naive schemes based on object detection or scene recognition are doomed to fail in these binary classification problems as illustrated in Figure 7.1.

7.3 Method

We now describe our supplementary method for predicting two types of human rights abuses based on people's emotional traits. Our goal is to label challenging everyday photos as either human-rights-abuse positive ('child labour' or 'displaced populations') or human-rights-abuse negative ('no child labour' or 'no displaced populations' respectively).

We first introduce a new mechanism capable of characterising an image based on two emotional states of all people in the scene, termed `global emotional traits (GET)`. This mechanism exploits two of the continuous dimensions of the VAD emotional state model

(a) children playing

(b) child labour

(c) camping

(d) displaced populations

Figure 7.1: In many cases, two of the most sought-after modern human rights abuses, `child labour` and `displaced populations`, cannot be identified only by properties of the surrounding scene and its related objects in a binary classification setting. This happens because a non-violent action like camping consists of specific objects (tents) that are resembling in camps with displaced populations or in the case of children playing in a non-urban area which can be visually very similar to the surrounding scene of child labour activity.

which are relevant to human rights image analysis. As will be explained in the following, global emotional traits are learned by jointly analysing each person and the entire scene. To detect GET of an image, we need to accurately localise the box containing a *human* and the associated object of interaction (denoted by b_h and b_o, respectively), as well as identify the emotional states e of each human using the VAD model. Our proposed solution adopts the RetinaNet [60] object detection framework followed by an additional *emotional traits* branch that estimates the continuous dimensions of each detected person and then determines the global emotional traits of the given image.

Specifically, given a set of candidate boxes, RetinaNet outputs a set of object boxes and a class label for each box. While the object detector can predict multiple class labels, our model is concerned only with the 'person' class. The region of the image comprising the person whose feelings are to be estimated at b_h is used alongside the entire image for simultaneously

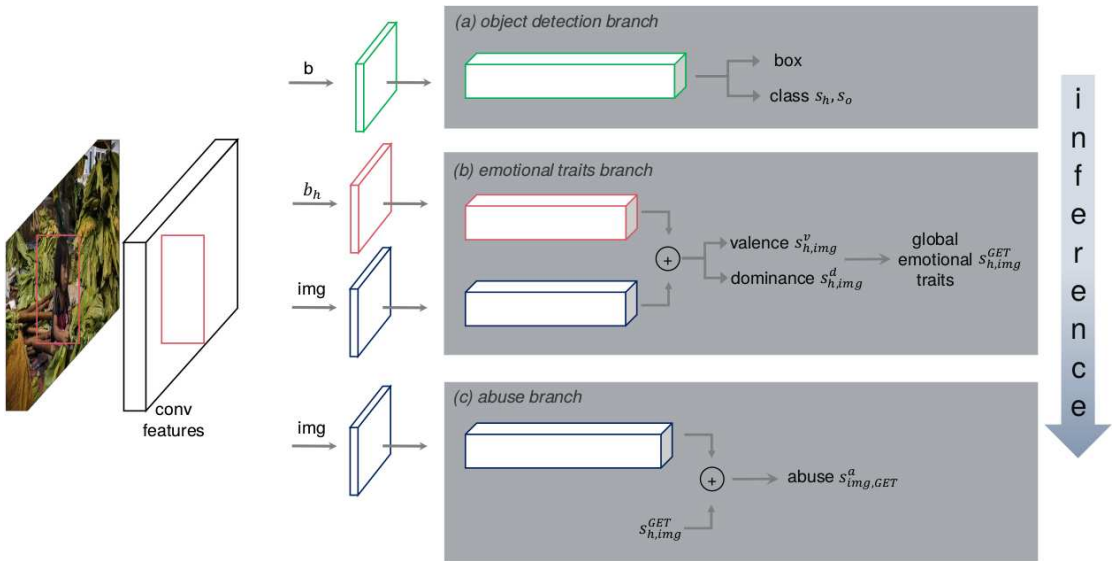


Figure 7.2: GET-AID model architecture. Our model consists of (a) an object detection branch, (b) an emotional traits branch, and (c) an abuse branch. The image features and their layers are shared between the emotional traits and abuse branches (blue boxes).

extracting their most relevant features. These features, are fused and used to perform continuous emotion recognition in VAD space. Our model extends typical image classification by assigning a triplet score $s^a_{img,GET}$ to pairs of candidate human boxes b_h and an abuse category a. To do so, we decompose the triplet score into three terms:

$$s^a_{img,GET} = s_h \cdot s^{GET}_{h,img} \cdot s^a_{img} \tag{7.1}$$

We discuss each component next, followed by details for training and inference. Figure 7.2 illustrates each component in our full framework.

7.3.1 Model components

Object detection branch. The *object detection* branch of our network, shown in Figure 7.2a, is identical to that of RetinaNet [60] single stage classifier. First, an image is forwarded through ResNet-50 [31], then in the subsequent pyramid layers, the more semantically important features are extracted and concatenated with the original features. For each proposal box b, we perform object classification and bounding-box regression to obtain a new set of boxes, each of which has an associated score s_o (or s_h if the box is assigned to the person category).

Emotional traits branch. The first role of the *emotional traits* branch is to assign a valence classification score s^v_{img} that measures how positive or pleasant an emotion is, ranging from negative to positive, and a dominance classification score s^d_{img} that measures the control level

of the situation by the person, ranging from submissive / non-control to dominant / in-control, to each human box b_h. Similar to [52], we use an end-to-end model with three main modules: two feature extractors and a fusion module. The first module takes the region of the image comprising the person whose emotional traits are to be estimated, b_h, while the second module takes as input the entire image, *img*, and extracts global features. This way the required contextual support is accommodated in the emotion recognition process. Finally, a third module takes as input the extracted image and body features and estimates the continuous dimensions in VAD space. The complete pipeline of the model is shown in Figure 6.5, while Figure 7.3 shows qualitative results of the method.

The second role of the *emotional traits* branch is to assign a two-dimensional emotion classification score $s^{GET}_{h,img}$ which characterises the entire input image, based on the two aforementioned emotion classification scores for each human. *GET* score, is the first attempt to establish a method for summarising the overall mood of a situation depicted in a single image. We decompose the two-dimensional GET score into two terms:

$$s^{GET}_{h,img} = s^{v}_{img} \cdot s^{d}_{img} \tag{7.2}$$

In the above, s^{v}_{img} is the encoding of the global valence score relative to human box b_h and entire image *img*, that is:

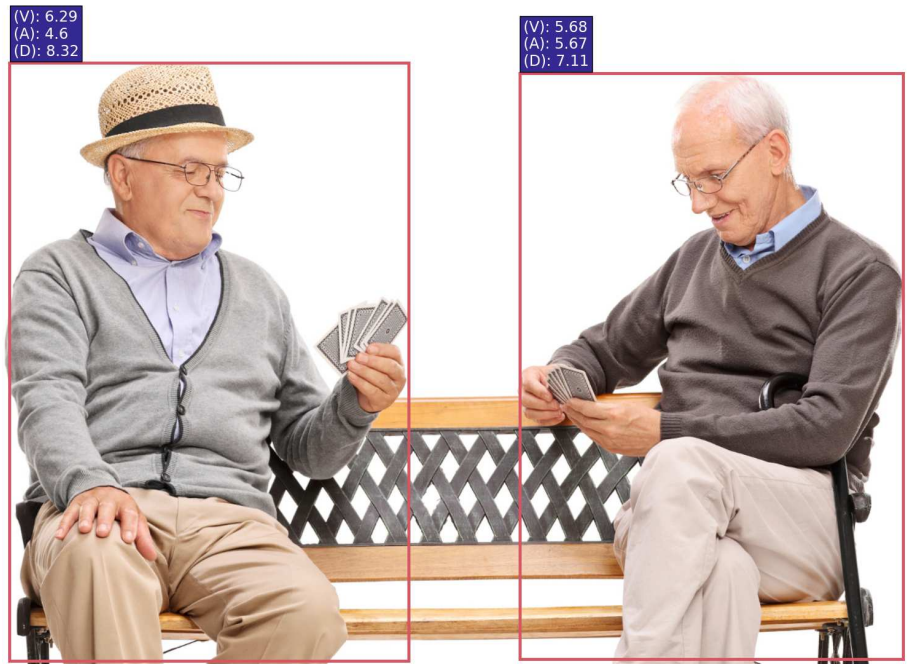
$$s^{v}_{img} = \frac{1}{n}\sum_{i=1}^{n} s^{v}_{h,img} \tag{7.3}$$

Similarly, s^{d}_{img} is the encoding of the global dominance score relative to human box b_h and entire image *img*, that is:

$$s^{d}_{img} = \frac{1}{n}\sum_{i=1}^{n} s^{d}_{h,img} \tag{7.4}$$

In Figure 7.4a and Figure 7.5a the three different emotional states over target objects location are estimated, while Figure 7.4b and 7.5b illustrate the GET proposed here. For the sake of completeness all three predicted numerical dimensions are depicted. However, only valence and dominance are considered to be relevant to the two human rights violation recognition scenarios since the agitation level of a person, denoted by arousal, can be ambiguous for several situations. For example, both Figure 7.4a and Figure 7.5a depict people with arousal values close to 5.5, but the activities captured are utterly different from a human rights perspective.

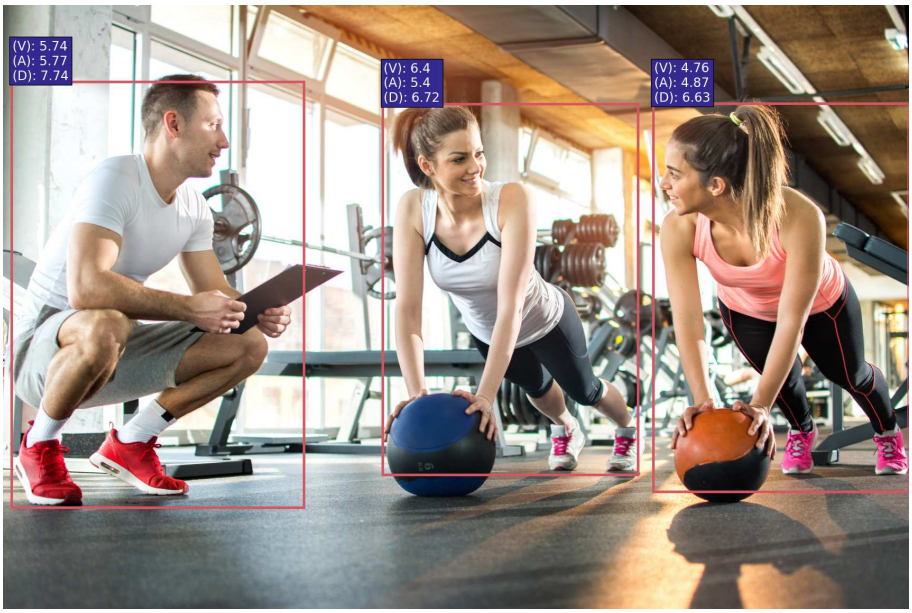
Abuse branch. The first role of the abuse branch, shown in Figure 7.2c, is to assign an abuse classification score to the input image. Just like in the two-phase transfer learning scheme deployed previously for HRA-CNNs in Chapter 4, we train an end-to-end model for



(a)

(b)

Figure 7.3: Continuous emotion recognition in VAD Space. Examples of people marked with the red bounding box that have been labelled with different scores of Valence, Arousal and Dominance.

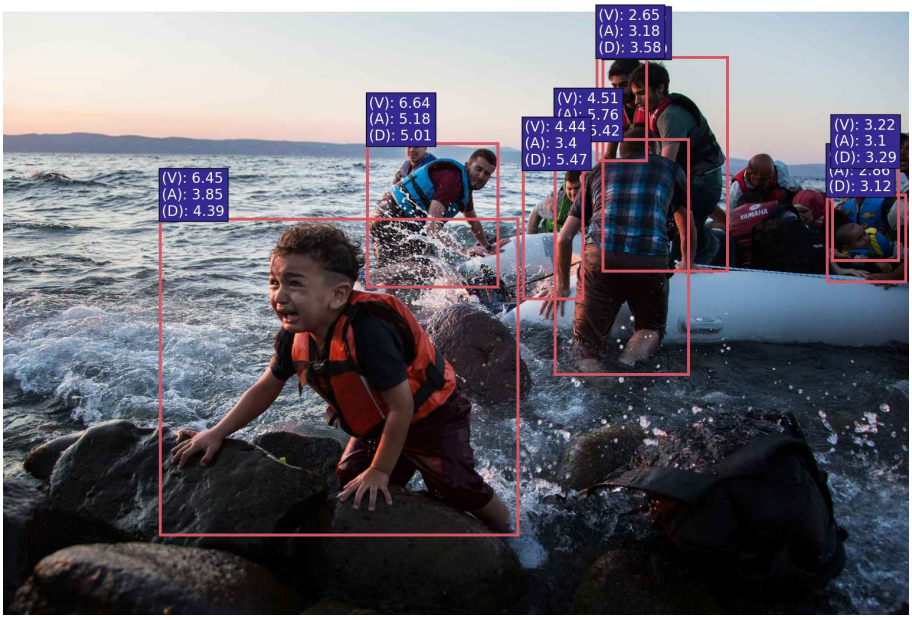


(a) Continuous emotion recognition in VAD space

(b) Our proposed *global emotional traits*

Figure 7.4: Example of estimating continuous emotions in VAD space vs our proposed global emotional traits (GET) from the combined body and image features. (a) shows the predicted emotional states and their scores from the person region of interest (RoI), while (b) shows the same image characterised by GET proposed here. The GET score will be later integrated with the standard image classification scores s^a_{img} to identify two types of abuses.

classifying everyday photos as either human-rights-abuse positive ('child labour' or 'displaced populations') or human-rights-abuse negative ('no child labour' or 'no displaced populations').



(a) Continuous emotion recognition in VAD space

(b) Our proposed *global emotional traits*

Figure 7.5: Further example of estimating continuous emotions in VAD space vs our proposed global emotional traits (GET) from the combined body and image features.

In order to improve the discriminative power of our model, the second role of the abuse branch is to integrate $s^{GET}_{h,img}$ in the recognition pipeline. Specifically, the raw image classification score s^{a}_{img} is readjusted based on the recognised global emotional traits. Each GET unit, that is deltas from the neutral state, is expressed as a numeric weight varying between 1 and 10. We

empirically set the numeric values of neutral emotional states between 4.5 and 5.5 using the number of examples per each of the scores in the continuous dimensions reported in [52]. The GET feature of each input image can be written in the form of a 2-element vector:

$$\bar{d} = (D_1, D_2) \tag{7.5}$$

where D_1 and D_2 refer to the weights of valence and dominance, respectively. The adjustment, adj, that will be assigned to the raw classification probability, s^a_{img} is the weight of valence/dominance multiplied by a factor of 0.11 which has been experimentally set. We treat ϕ as a hyperparameter that we empirically set to $\phi = 0.11$ using the EMOTIC test set. When the input image depicts positive valence or positive dominance, the adjustment factor is subtracted from the positive human-rights-abuse probability and added to the negative human-rights-abuse probability. Similarly, when the input image depicts negative valence or negative dominance the adjustment factor is added to the negative human-rights-abuse probability and subtracted from the positive human-rights-abuse probability. This is formally written in Algorithm 2. Finally, when no b_h were detected from the object detection branch, (7.1) is reduced into plain image classification as follows:

$$s^a_{img,GET} = s^a_{img} \tag{7.6}$$

7.3.2 Training

We approach human rights abuse classification as a *cascaded, multi-task* learning problem. Due to different datasets, convergence times and loss imbalance, all three branches have been trained separately. For object detection we adopted an existing implementation of the RetinaNet object detector, pre-trained on the COCO dataset [61], with a ResNet-50 backbone. Specifically, RetinaNet-50 achieves 32.5 accuracy (AP) with a speed of 73ms on COCO *test-dev*.

For emotion recognition in continuous dimensions, we formulate this task as a regression problem using the Euclidean loss. The two feature extraction modules described in *emotional traits recognition* section, are designed as truncated versions of various well-known CNNs and initialised using pretrained models on two large-scale image classification datasets, ImageNet [54] and Places [119]. The truncated version of those CNNs removes the fully connected layer and outputs features from the last convolutional layer in order to maintain the localisation of different parts of the images which is significant for the task at hand. Features extracted from these two modules (red and blue boxes in Figure 7.2b) are then combined by a fusion module. This module first uses a global average pooling layer to reduce the number of features from each network and then a fully connected layer, with an output of a 256-D vector, functions

Algorithm 2: Calculate $s^{a}_{img,GET}$

Require: $b_h > 0$
$s_{pos} \leftarrow s^{v}_{img}$ {v: human-rights-abuse positive}
$s_{neg} \leftarrow s^{nv}_{img}$ {nv: human-rights-abuse negative}
$\phi \leftarrow 0.11$
if $D_1 \geq 4.5$ **and** $D_1 \leq 5.5$ **then**
  $s_{pos} = s^{pos}_{img}$
  $s_{neg} = s^{neg}_{img}$
else if $D_1 > 5.5$ **then**
  $adj = (D_1 - 5.5) * \phi$
  $s_{pos} = s_{pos} - adj$
  $s_{neg} = s_{neg} + adj$
else if $D_1 < 4.5$ **then**
  $adj = (4.5 - D_1) * \phi$
  $s_{pos} = s_{pos} + adj$
  $s_{neg} = s_{neg} - adj$
end if
if $D_2 \geq 4.5$ **and** $D_2 \leq 5.5$ **then**
  Return s_{pos}, s_{neg}
else if $D_2 > 5.5$ **then**
  $adj = (D_2 - 5.5) * \phi$
  $s_{pos} = s_{pos} - adj$
  $s_{neg} = s_{neg} + adj$
else if $D_2 < 4.5$ **then**
  $adj = (4.5 - D_2) * \phi$
  $s_{pos} = s_{pos} + adj$
  $s_{neg} = s_{neg} - adj$
end if
Return s_{pos}, s_{neg}

as a dimensionality reduction layer for the concatenated pooled features. Finally, we include a second fully connected layer with 3 neurons representing valence, arousal and dominance. The parameters of the three modules are learned jointly using stochastic gradient descent with momentum of 0.9. The batch size is set to 54 and we use dropout with a ratio of 0.5.

For human rights abuse classification, we formulate this task as a binary classification problem. We train an end-to-end model for classifying everyday images as human-rights-abuse positive or human-rights-abuse negative, based on the context of the images, for two independent scenarios, namely *child labour* and *displaced populations*. Following the two-phase transfer learning scheme proposed in 4.3, we fine-tune various CNN models for the two-class classification task. First, we conduct feature extraction utilising only the convolutional

Body Feature Backbone	Mean error rate
VGG16	1.59
VGG19	1.57
ResNet50	1.69
VGG16 + ResNet50	1.40
VGG16 + VGG19	1.36
VGG19 + ResNet50	1.48
VGG19 + ResNet50 + VGG16	1.36

Table 7.1: Emotion recognition results using the continuous dimensions emotion representation in the form of mean error rate (average of all three VAD dimensions) for different body feature backbone CNNs. The image feature backbone CNN was kept constant for all cases, namely VGG16-Places365 [119].

base of the original networks in order to end up with more generic representations as well as retaining spatial information similar to emotion recognition pipeline. The second phase consists of unfreezing some of the top layers of the convolutional base and jointly training a newly added fully connected layer and these top layers. All the CNNs presented here [1] were trained using the Keras Python deep learning framework [8] over TensorFlow [1] on Nvidia GPU P100.

7.3.3 Inference

Object Detection Branch: We first detect all objects (including the person class) in the input image. We apply a threshold on boxes with scores higher than 0.5, which is set conservatively to retain most objects. This yields a set of n boxes b with scores s_h and s_o. These boxes are used as input to the emotional trait branch.

Emotional Traits Branch: Next, we apply the emotional traits branch to all detected objects that were classified as *human*. We feed each human box b_h alongside the entire input image *img* to the VAD emotion recognition model. For each b_h, we predict valence s^v_{img} and arousal s^a_{img} scores, and then compute the *global emotional traits* $s^{GET}_{h,img}$ that describes the entire input image *img*.

Abuse Branch: If no human box b_h has been detected, for example when a plain beach without people or kitchen appliances was given as input image, the branch predicts the two abuse scores s_v and s_{nv} directly from the binary classifier. On the other hand, when one or more people have been detected, the branch weights the raw predictions from the binary classifier based on the computed global emotional traits of the input image according to Algorithm 2.

[1] Available at https://github.com/GKalliatakis/GET-AID

7.4 Implementation Details

Our emotion recognition implementation is based on the emotion recognition in context (EMOTIC) model [52], with the difference that our model estimates only continuous dimensions in VAD space. We train the three main modules on the EMOTIC database, which contains a total number of 18,316 images with 23,788 annotated people, using pre-trained CNN feature extraction modules. We treat this multiclass-multilabel problem as a regression problem by using a weighted Euclidean loss to compensate for the class imbalance of EMOTIC.

We evaluate the continuous dimensions using error rates - the difference (in average) between the true value and the regressed value. Table 7.1 shows the results for the continuous dimensions using error rates. The best result is obtained by utilising *model ensembling*, which consists of pooling together the predictions of a set of different models in order to produce better predictions. We pool the predictions of classifiers (*ensemble the classifiers*) by conducting weighted average of their prediction at inference time. The weights are learned on the validation data - usually the better single classifiers are assigned with a higher weight, while the worst single classifiers are assigned a lower weight. However, ensembling the classifiers results in prolonged inference times, which causes us to turn our focus onto single classifiers for the remainder of the experiments.

Following the two-phase transfer learning scheme proposed in Chapter 4, we fine-tune our human-rights-abuse classification models, Figure 7.2 (c), for 50 iterations on the HRA—Binary *trainval* set, introduced in section 3.3.5, with a learning rate of 0.0001 using the stochastic gradient descent (SGD) [55] optimizer for cross-entropy minimization. These *vanilla* models will be examined against GET-AID. Here, vanilla means pure image classification using solely fine-tuning without any alteration. Also, for the displaced populations scenario, GET-AID will be compared with DisplaceNet presented in Chapter 6.

7.5 Quantitative Results

To enable a fair comparison between vanilla CNNs[2] and GET-AID, we use the same backbone combinations for all modules described in Figure 7.2. We report comparisons in both *top-1 accuracy* and *coverage* metrics for fine-tuning up to three convolutional layers. The per-network results for *child labour* and *displaced populations* are shown in Table 7.2 and Table 7.3 respectively.

Vanilla CNNs. The implementation of vanilla CNNs is solid: it has up to 67% top-1 accuracy on the child labour classification and up to 82% top-1 accuracy on the displaced populations

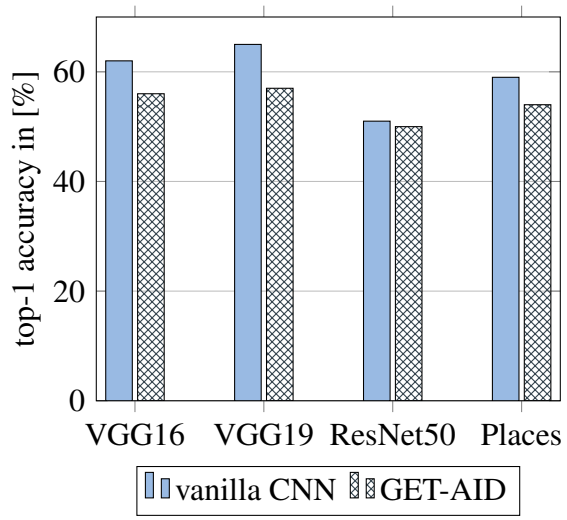
[2]Here, vanilla means pure image classification using solely fine-tuning without any alteration

backbone CNN	layers fine-tuned	vanilla CNN			GET-AID		
		Top-1 acc.	Coverage	Weighted Sum	Top-1 acc.	Coverage	Weighted Sum
VGG16	1	62%	73%	33.75	56%	**78%**	33.5
VGG19		65%	30%	23.75	57%	**55%**	28
ResNet50		51%	0%	12.75	50%	**24%**	18.5
Places365		59%	71%	32.5	54%	**81%**	33.75
VGG16	2	61%	77%	34.5	59%	**78%**	34.25
VGG19		61%	64%	31.25	59%	**76%**	33.75
ResNet50		52%	0%	13	49%	**33%**	20.5
Places365		54%	44%	24.5	52%	**65%**	29.25
VGG16	3	56%	83%	34.75	56%	**84%**	35
VGG19		55%	**87%**	35.5	55%	82%	34.25
ResNet50		50%	**99%**	**37.25**	48%	91%	34.75
Places365		67%	0%	16.75	53%	**30%**	20.75
mean	-	58%	52.33%	27.58	54%	**64.75%**	**29.68**

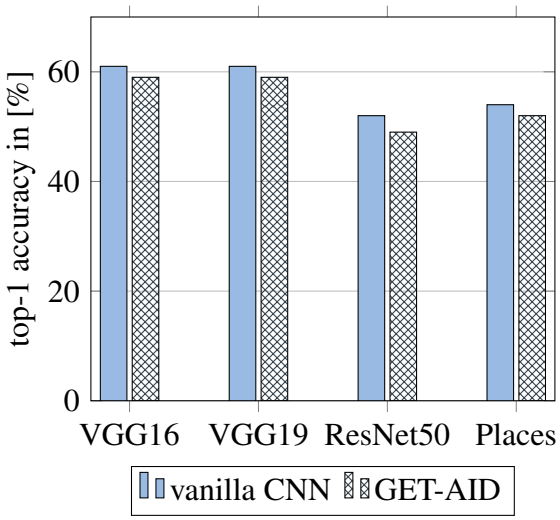
Table 7.2: Top-1 accuracy and coverage obtained on test set of HRA—Binary for the *child labour* scenario using GET-AID. We show the main baseline and GET-AID for various network backbones. We bold the leading entries on coverage.

backbone CNN	layers fine-tuned	vanilla CNN			GET-AID		
		Top-1 acc.	Coverage	Weighted Sum	Top-1 acc.	Coverage	Weighted Sum
VGG16	1	58%	0%	14.5	56%	**24.4%**	20.1
VGG19		69%	3%	18	59%	**33%**	23
ResNet50		60%	0%	15	53%	**29%**	20.5
Places365		64%	3%	16.75	54%	**32%**	21.5
VGG16	2	63%	43%	26.5	60%	**58%**	29.5
VGG19		77%	54%	32.75	70%	**59%**	32.25
ResNet50		42%	1%	10.75	44%	**33%**	19.25
Places365		80%	49%	32.25	73%	**58%**	32.75
VGG16	3	72%	69%	35.25	67%	**71%**	34.5
VGG19		82%	64%	36.5	77%	**68%**	**36.25**
ResNet50		53%	0%	13.25	51%	**22%**	18.25
Places365		81%	66%	36.75	71%	66%	34.25
mean	-	66.75%	29.37%	24.03	61.25%	**46.11%**	**26.84**

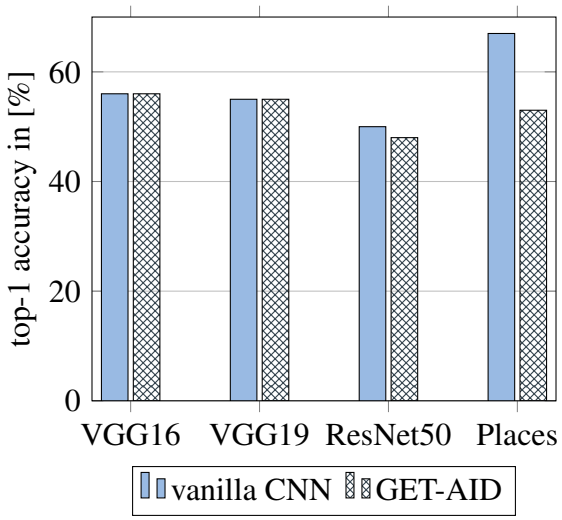
Table 7.3: Top-1 accuracy and coverage obtained on test set of HRA—Binary for the *displaced populations* scenario using GET-AID. We show the main baseline and GET-AID for various network backbones. We bold the leading entries on coverage.



(a) 1 layer fine-tuned



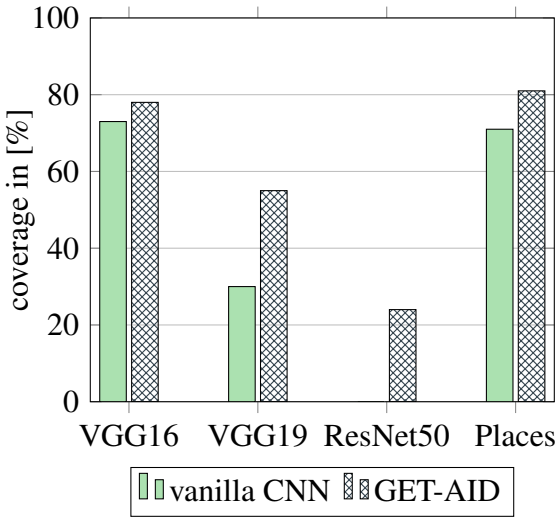
(b) 2 layers fine-tuned



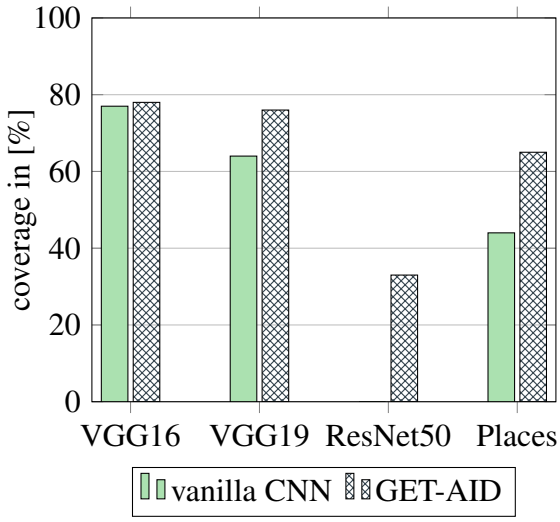
(c) 3 layers fine-tuned

Figure 7.6: Comparative results in terms of top-1 accuracy for fine-tuned models (vanilla CNN) and our proposed method, *GET-AID*, over various backbone networks for the `child labour` scenario.

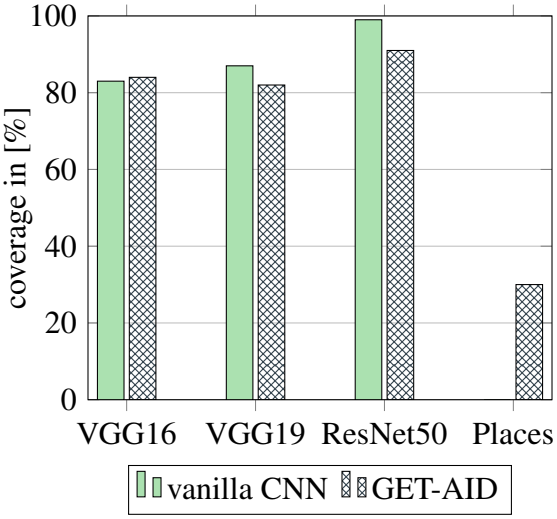
classification. That is 29.82 and 46.82 points higher than the best performing *multiclass classification* HRA-CNNs presented in section 4.3, which achieved an accuracy of 35.18% across all 9 categories. Note that only cases with a single or two layers fine-tuned were considered for calculating these numbers in order to be consistent with the implementation of *HRA CNNs* reported in Chapter 4. Regarding coverage, vanilla CNNs achieve up to 99% for child labour classification and up to 69% child displaced populations classification, which is in



(a) 1 layer fine-tuned



(b) 2 layers fine-tuned

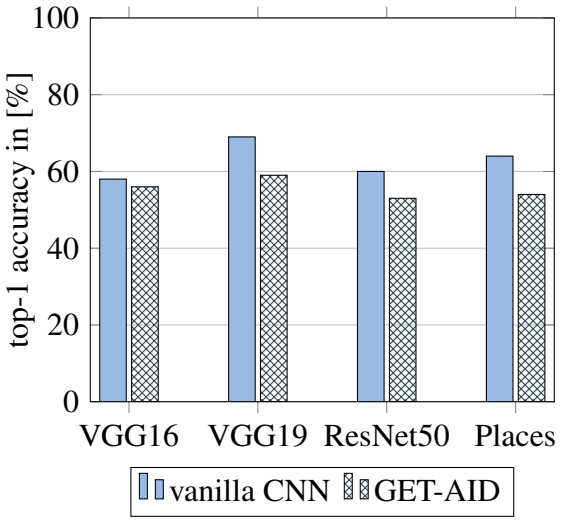


(c) 3 layers fine-tuned

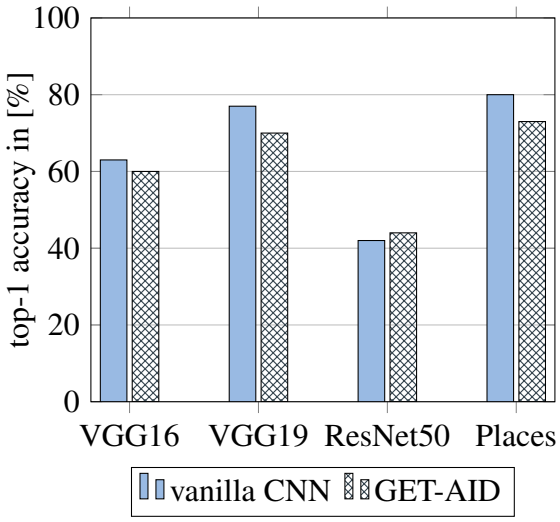
Figure 7.7: Comparative results in terms of coverage for fine-tuned models (vanilla CNN) and our proposed method, *GET-AID*, over various backbone networks for the `child labour` scenario.

par with the 64% maximum coverage reported in Chapter 4. We believe that this accuracy gap is mainly due to the fact that Human Rights Archive CNNs deal with a multiclass classification problem, whereas in this work we classify inputs into two mutually exclusive classes.

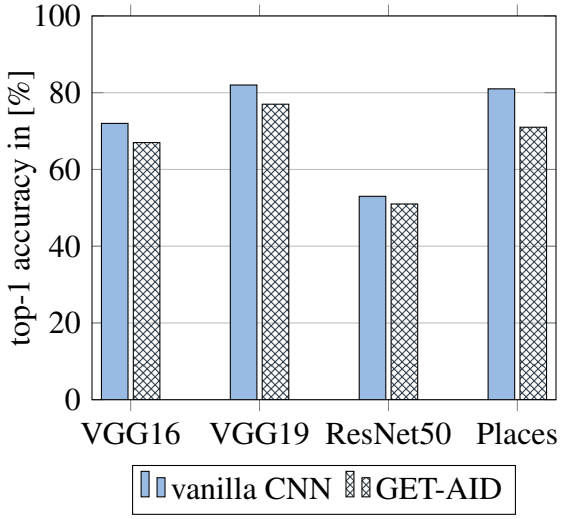
Scenario 1: Child labour. GET-AID obtains a mean accuracy of 54%, which is marginal decrease over the strong baselines of 58% achieved by vanilla CNNs. In relation to coverage, GET-AID, achieves a mean coverage of 64.75% on the HRA—Binary *test* set. This is a



(a) 1 layer fine-tuned



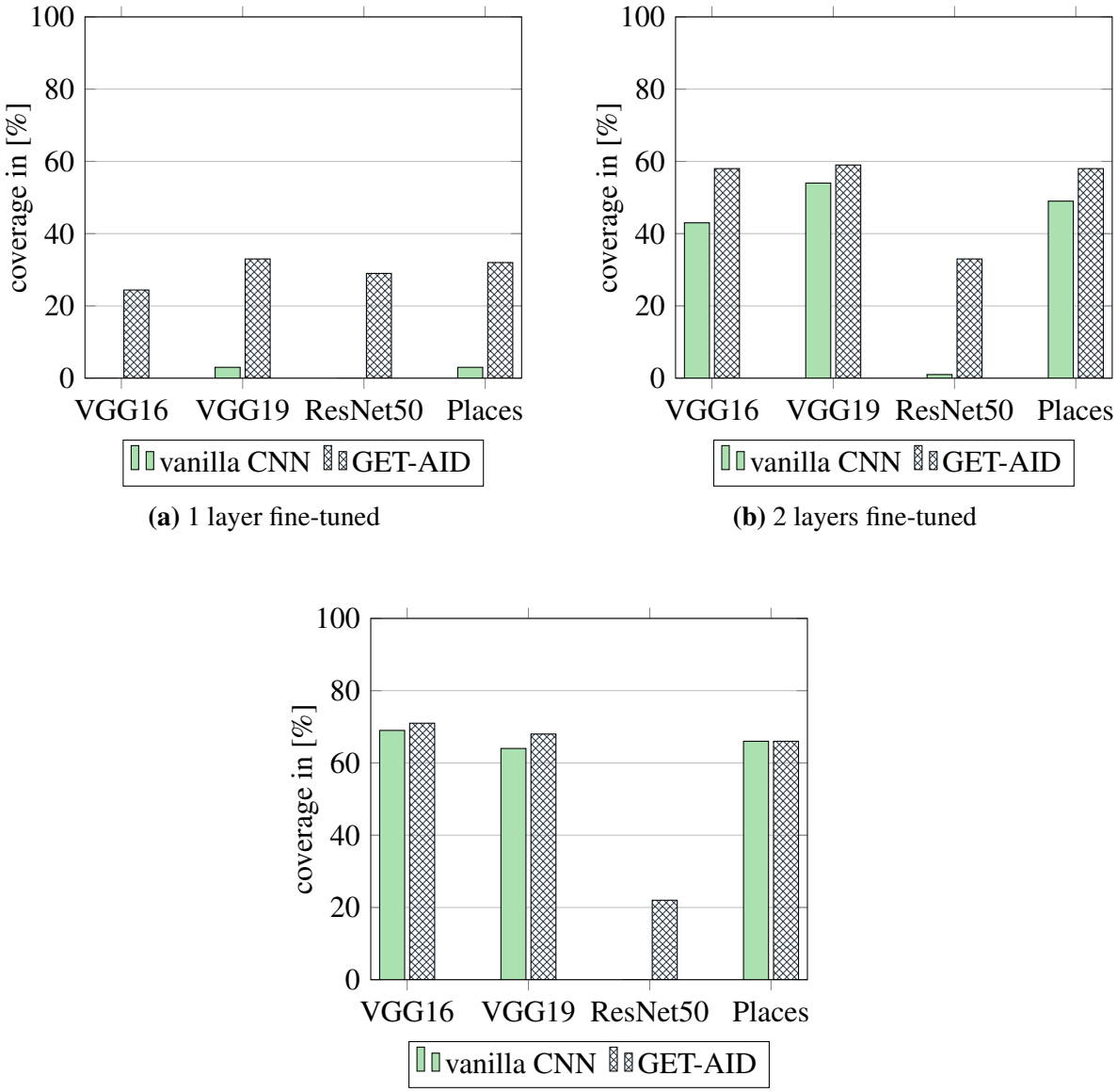
(b) 2 layers fine-tuned



(c) 3 layers fine-tuned

Figure 7.8: Comparative results in terms of top-1 accuracy for fine-tuned models (vanilla CNN) and our proposed method, *GET-AID*, over various backbone networks for the `displaced populations` scenario.

significant increase of 12.42 points over the strong baselines of 52.33% achieved by vanilla CNNs. This indicates a relative gain of 23.73%. Comparative results for all three fine-tuned layers are illustrated in Figure 7.6 and Figure 7.7 respectively. It is evident that GET-AID constantly improves the coverage performance of the system, even for extreme cases where vanilla CNNs fail to produce robust predictions at all (ResNet50 in Figure 7.7a and Figure 7.7b,



(a) 1 layer fine-tuned

(b) 2 layers fine-tuned

(c) 3 layers fine-tuned

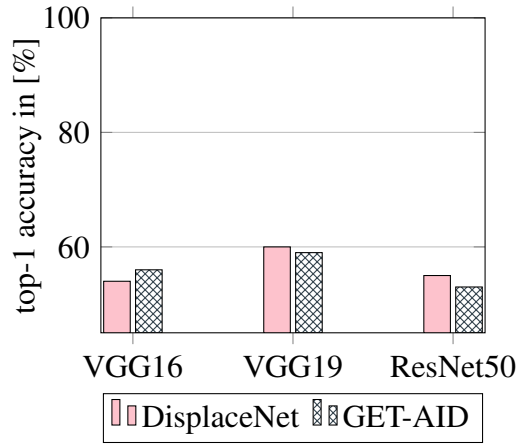
Figure 7.9: Comparative results in terms of coverage for fine-tuned models (vanilla CNN) and our proposed method, *GET-AID*, over various backbone networks for the `displaced populations` scenario.

and Places in Figure 7.7c). In order to reach this level of coverage, GET-AID sacrifices top-1 accuracy for all instances only by a small margin as seen in Figure 7.6.

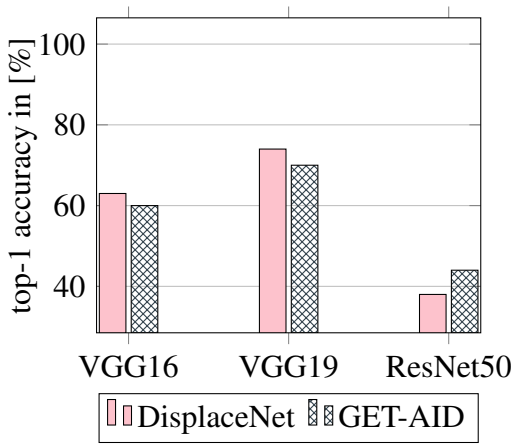
Scenario 2: Displaced populations. GET-AID, achieves a mean coverage of 46.11% on the HRA—Binary *test* set. This is an absolute gain of 16.74 points over the strong baselines of 29.37% achieved by vanilla CNNs. This indicates a relative improvement of 57.21%. In relation to accuracy, GET-AID obtains a mean accuracy of 61.25%, which is marginal decrease

backbone CNN	layers fine-tuned	DisplaceNet			GET-AID		
		Top-1 acc.	Coverage	Weighted Sum	Top-1 acc.	Coverage	Weighted Sum
VGG16	1	54%	3%	14.25	56%	**24.4%**	20.1
VGG19		60%	6%	16.5	59%	**33%**	23
ResNet50		55%	4%	14.75	53%	**29%**	20.5
VGG16	2	63%	49%	28	60%	**58%**	29.5
VGG19		74%	**68%**	**35.5**	70%	59%	32.25
ResNet50		38%	5%	10.75	44%	**33%**	19.25
mean	-	57.33%	20.83%	19.54	57%	**39.4%**	**24.1**

Table 7.4: Top-1 accuracy and coverage obtained for the displaced populations scenario using GET-AID and DisplaceNet. GET-AID achieves higher coverage by a considerable margin of 18.57 points. Note, that we show results only for the settings that are common amongst the two approaches. We bold the leading entries on coverage.



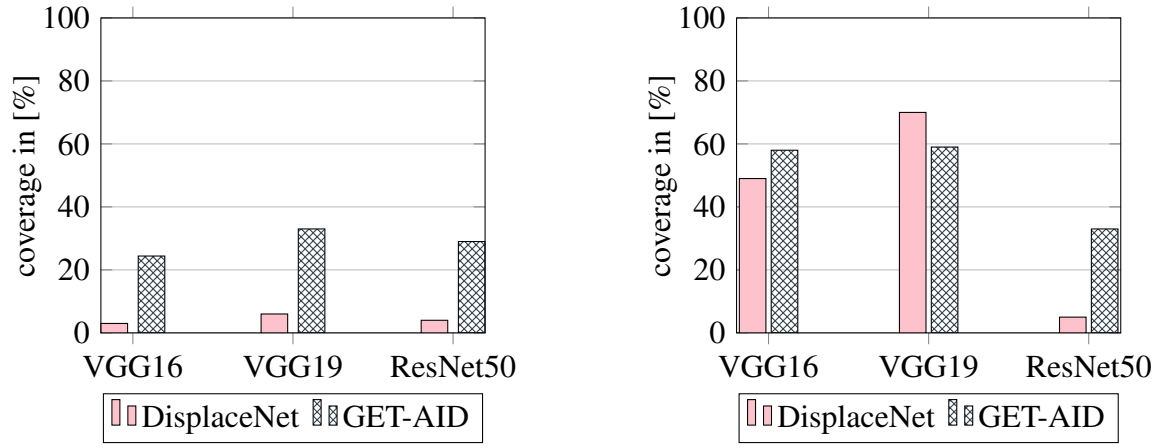
(a) Top-1 accuracy on the test set of HRA–Binary with one layer of the backbone network fine-tuned.



(b) Top-1 accuracy on the test set of HRA–Binary with two layers of the backbone network fine-tuned.

Figure 7.10: Comparative results in terms of top-1 accuracy for the `displaced populations` scenario using GET-AID and DisplaceNet over various backbone networks.

over the strong baselines of 66.75% achieved by vanilla CNNs. Comparative results for all three fine-tuned layers are illustrated in Figure 7.8 and Figure 7.9. Results follow the same pattern as in scenario 1. GET-AID constantly improves the coverage performance of the system, even for extreme cases where vanilla CNNs fail to produce robust predictions at all (VGG16, ResNet50 in Figure 7.9a and ResNet50 in Figure 7.9c). Again, in order to reach this level of coverage, GET-AID sacrifices top-1 accuracy for all instances only by a small margin as seen in Figure 7.8.



(a) Coverage on the test set of HRA–Binary with one layer of the backbone network fine-tuned.

(b) Coverage on the test set of HRA–Binary with two layers of the backbone network fine-tuned.

Figure 7.11: Comparative results in terms of coverage for the `displaced populations` scenario using GET-AID and DisplaceNet over various backbone networks.

For this scenario, we further compare GET-AID and DisplaceNet (Chapter 6) on the same subset of HRA—Binary *test*, to examine whether two emotional traits can further improve the robustness of the predictions made by our system. DisplaceNet achieves a mean coverage of 20.83%, while GET-AID significantly improves mean coverage by 18.57 points, achieving 39.4%. Again, for a fair measurement, only two fine-tuned layers were included in the results presented in Table 7.4. Comparative results for the two fine-tuned layers for top-1 accuracy and coverage are illustrated in Figure 7.10 and Figure 7.11 respectively.

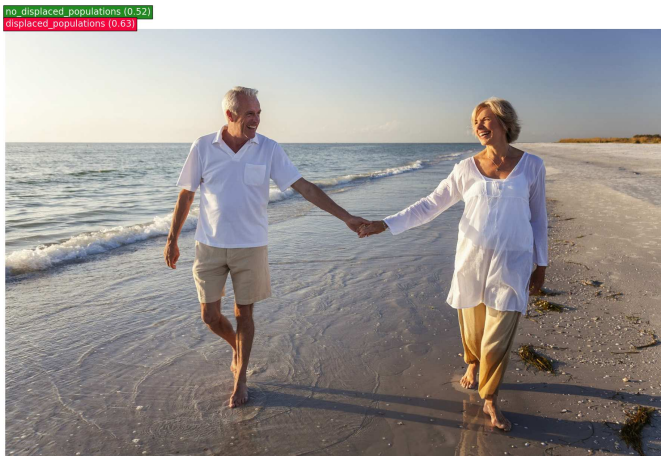
We believe that the negligible drop in top-1 accuracy for both scenarios is mainly due to the fact that the HRA—Binary *test* set is not solely made up of images with people in their context, it also contains images of generic objects and scenes, where only the sole classifier's prediction is taken into account.

7.6 Qualitative Results

We show our human rights abuse detection results in Figure 7.12 and Figure 7.13. Each subplot illustrates two predictions alongside their probability scores. The top of the two predictions is given by GET-AID, while the bottom one is given by the respective vanilla CNN sole classifier. Our method can successfully classify human rights abuses by overturning the initial-false prediction of the vanilla CNN as shown in Figure 7.12. Moreover, GET-AID can strengthen the initial-true prediction of the sole classifier as shown in Figure 7.13.

(a)

(b)



(c)

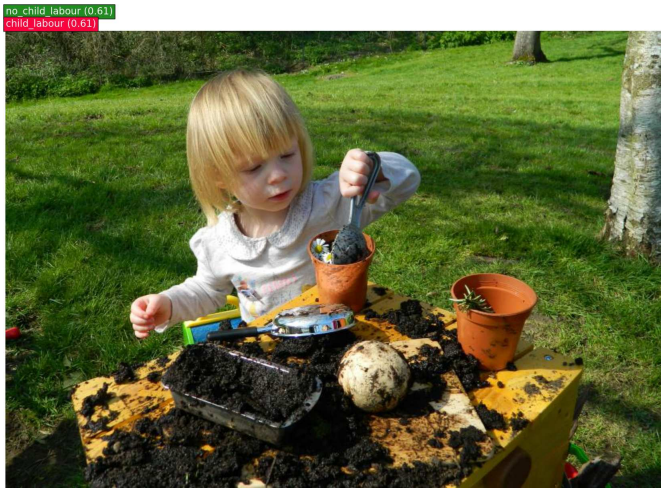
Figure 7.12: Human rights abuses detected by GET-AID for the `displaced populations` scenario.

7.6.1 Failure Cases

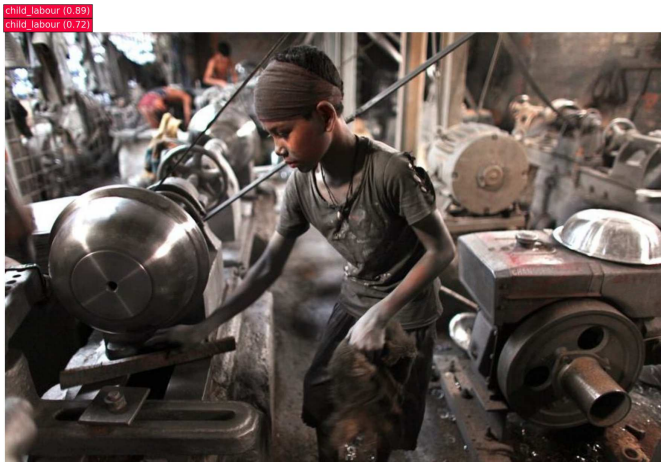
Figure 7.14 shows some failure detections. Our method can be incorrect, because of false global emotional traits inferences. Some of them are caused by a failure of continuous dimensions emotion recognition, which is an interesting open problem for future research.

7.7 Summary

In this chapter we have presented a remodelled, human-centric approach for classifying two types of human rights abuses from everyday, challenging photos. This two-class human rights abuse labelling problem is not trivial, given the high-level image interpretation required. As shown in Chapter 6, emotion perception based on people's frame of reference is closely related with situations where human rights are being violated. In this chapter, we have introduced a novel mechanism, termed *global emotional traits*, which is responsible for weighting the classifiers prediction during inference, based on all people's emotional traits. We benchmark performance of our GET-AID model against sole CNN classifiers as well as DisplaceNet. Experimental results of two diverse scenarios, have showed that global emotional traits are a powerful strategy for analysing and interpreting human rights related imagery.



(a)



(b)

(c)

Figure 7.13: Human rights abuses detected by GET-AID for the `child labour` scenario.

(a)

(b)

Figure 7.14: False detections of GET-AID. For (a), GET-AID overturns the initial-true prediction of the vanilla CNN, while for (b) GET-AID strengthens the initial-false prediction.

Chapter 8

Conclusion

We conclude this thesis by providing a summary of our achievements and impact as a result of this work. We also outline some potential future directions for building on this work.

8.1 Achievements and Impact

Images bring human rights violations to life in a way that mere description and texts cannot. The use of technologies like computer vision can mitigate inequalities in the human rights domain. However, the approach of human rights practitioners to technology will be a determining factor in their ability to advance accountability, transparency, and justice in the years to come. This thesis is an effort to conceptualise the future of the intersection of computer vision and human rights practice by establishing the frontier of automated visual recognition of human rights violations along with gaining a deeper understanding of how this unexplored level of recognition can be accomplished in the wild.

In this thesis we have examined the domain shift problem of learning from images of everyday objects/scenes and applying this knowledge to real-world imagery in the context of human rights. We have also contributed several datasets for further research in the field, and have built a web demo allowing our research to be directly utilised. A brief account of the major contributions of this thesis is given below.

In Chapter 3, we introduce the first ever image datasets of human rights violations. This data has formed the basis of our research in Chapters 4, 5, 6 and 7, and has led to the release of three datasets, `HRUN`, `HRA`, and `HRA-Binary`. We also described the various acquisition procedures and demonstrated the importance of verified imagery in the context of human rights. We have publicly released all three datasets used in this thesis, so researchers may use this as a benchmark for evaluating their classifier's performance on human rights violations.

In Chapter 4 we contributed a thorough examination of the transfer learning approach for applying natural image-trained classifiers to human rights context. We examined this for features produced by various neural network architectures both for linear SVM classifier as well as end-to-end classifiers. As small-scale dataset was seen to suffer from overfitting, we fine-tuned end-to-end models on more data utilising a novel two-phase technique to develop the first ever image classification benchmark in the context of human rights.

Chapter 5 showed that the performance of the trained classifiers of Chapter 4 on the `HRA dataset` could not be substantially improved by combining features extracted from different object-centric CNN models. We also combined deep features from models pretrained on objects and scenes in order to learn complementary cues. Interestingly, various combinations of these two pretrained networks was not seen to perform better than the sum of its parts, illustrating that objects or/and scenes are just a first step for machines to interpret human rights violations in the visual world. In this chapter we also presented a practical application of this research. We developed a web demo for predicting human rights violations that may be used by human rights advocates and analysts.

We then moved away from multi-class classification of human rights violations to explore the task of recognising displaced people from images. Displacement of people is a form of social change that results in millions of men, women and children to leave their homes every year, making it one of the most crucial issues that human rights investigators are targeting. In Chapter 6 we developed an emotion-based approach to infer potential displaced people from images by integrating their dominance level and CNN classifiers into one framework. One impact of this work was the improvement in coverage–the proportion of a data set for which a classifier is able to produce a prediction–of our fine-tuned CNNs by 4%. Furthermore, this method revealed that understanding people's emotional states from their frame of reference can be closely related with situations where human rights are being violated.

In Chapter 7 we build on findings from Chapter 6, and we develop a human-centric approach that exploits two powerful cues–how positive or pleasant an emotion is, and the control level of the situation–in order to recognise two of the most sought-after modern human rights violations, *child labour* and *displaced populations*. To achieve this, we introduce a new mechanism capable of characterising an input image based on the emotional states of all people in the scene, termed *global emotional traits (GET)*. We showed that the proposed *GET-AID* system further improves the coverage up to 23.73% for child labour and 57.21% for displaced populations. Furthermore, we have designed GET in a generic way, making it a plug-and-play unit without the need of changing network architectures or requiring hyperparameters tuning.

Finally, it is important to remember that the technological systems described in this thesis are best thought of as tools–they cannot document human rights violations on their own, but can increase the efficiency and effectiveness of human analysts when used properly.

8.2 Extensions and Future Work

We believe this work opens up more questions and avenues to explore than it closes off. There is much potential for automated visual recognition of human rights violations. Here, we discuss possibilities for future research in this context.

Improving image datasets. While our image datasets are of sufficient realism to enable generalisation of representation learning methods, they can be improved further. An interesting extension to consider is *natural image synthesis*. Despite recent progress in generative image modelling, successfully generating high-resolution, diverse samples from complex datasets such as HRA remains an elusive goal for the computer vision community.

Interrelation of HRA classes. As seen in Chapter 3, some classes in the HRA dataset are inevitably associated. For example child labour may result from children being left out of school or the other way around when being out of school forces children to work. Another example can be found in displaced populations and environment, where the latter most of the times plays a key role in people's movement. Capturing all those interrelationships could potentially benefit the recognition process. For this, it would perhaps be beneficial to describe classes using interpretable feature representations that incorporate attributes.

Learning a network from scratch with human rights violations. The CNNs used in this thesis have been pre-trained using natural images of objects or scenes, largely due to a lack of annotation in the context of human rights. Therefore, it would be interesting to explore data generations techniques to construct a large-scale image dataset that potentially could be utilised to train a network from scratch.

Action recognition in human rights context. Future work on human rights violation recognition could explore whether this task can be improved by utilising a network that can recognise different actions or models capable of recognising human-object interactions.

Story-telling of an image. For most of us, a picture can be interpreted as a rich amount of semantically meaningful information. This kind of semantic interpretation of the visual world is called high-level visual recognition. This is one of the most fundamental and important functionalities of an intelligence system which is required for improving the effectiveness of HRVR. Much work remains to be done on this field within from the computer vision community.

Multimodal integration. Another interesting direction for future research could be information integration from different systems such as video, and text. However, this presents some difficulties: (i) there are still far fewer videos than still images typically used to train a network; (ii) training and deploying algorithms trained on videos will be much more intensive in terms of computational power; (iii) we cannot assume complete annotation for text reports; (iii) there is a serious imbalance in the number of languages used for reporting human rights violations.

Bibliography

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA*, volume 16, pages 265–283.

[2] Agrawal, P., Girshick, R., and Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 329–344. Springer International Publishing.

[3] Aronson, J. D. (2018). Computer vision and machine learning for human rights video analysis: Case studies, possibilities, concerns, and limitations. *Law & Social Inquiry*, 43(4):1188–1209.

[4] Aronson, J. D., Xu, S., and Hauptmann, A. (2015). Video analytics for conflict monitoring and human rights documentation. *Center for Human Fights Science Technical Report*.

[5] Azizpour, H., Razavian, A. S., Sullivan, J., Maki, A., and Carlsson, S. (2016). Factors of transferability for a generic convnet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1790–1802.

[6] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

[7] Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., and Yuille, A. (2014). Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1971–1978. IEEE Computer Society.

[8] Chollet, F. et al. (2015). Keras. Available from: https://keras.io. Accessed: 15 January 2019.

[9] Chu, W.-S., De la Torre, F., and Cohn, J. F. (2016). Selective transfer machine for personalized facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):529–545.

[10] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.

[11] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893. IEEE Computer Society.

[12] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE Computer Society.

[13] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the International Conference on Machine Learning*, pages 647–655.

[14] Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666.

[15] Dubberley, S., Griffin, E., and Bal, H. M. (2015). Making secondary trauma a primary issue: A study of eyewitness media and vicarious trauma on the digital frontline. *Eyewitness Media Hub*.

[16] Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124.

[17] Eleftheriadis, S., Rudovic, O., and Pantic, M. (2016). Joint facial action unit detection and feature fusion: A multi-conditional learning approach. *IEEE transactions on image processing*, 25(12):5727–5742.

[18] Ergun, H., Akyuz, Y. C., Sert, M., and Liu, J. (2016). Early and late level fusion of deep convolutional neural networks for visual concept recognition. *International Journal of Semantic Computing*, 10(03):379–397.

[19] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338.

[20] Fabian Benitez-Quiroz, C., Srinivasan, R., and Martinez, A. M. (2016). Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5562–5570. IEEE Computer Society.

[21] Friesen, E. and Ekman, P. (1978). Facial action coding system: a technique for the measurement of facial movement. *Palo Alto*, 3.

[22] Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., and Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*.

[23] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

[24] Garvie, C. (2016). *The perpetual line-up: Unregulated police face recognition in America*. Georgetown Law, Center on Privacy & Technology.

[25] Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448.

[26] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587. IEEE Computer Society.

[27] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158.

[28] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[29] Hassan, U., Ali, B., and Shelina, J. (2016). ARCADE: ARtillery Crater Analysis and Detection Engine. Available from: https://rudiment.info/project/arcade/. Accessed: 18 January 2019.

[30] He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.

[31] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE Computer Society.

[32] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708. IEEE Computer Society.

[33] Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154.

[34] Huh, M., Agrawal, P., and Efros, A. A. (2016). What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614*.

[35] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311. IEEE Computer Society.

[36] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM.

[37] Juefei-Xu, F., Luu, K., and Savvides, M. (2015). Spartans: single-sample periocular-based alignment-robust recognition technique applied to non-frontal scenarios. *IEEE Transactions on Image Processing*, 24(12):4780–4795.

[38] Kalliatakis, G. (2018). Human Rights UNderstanding CNNs. https://github.com/GKalliatakis/Human-Rights-UNderstanding-CNNs/releases/tag/v1.0. Accessed: 14 May 2019.

[39] Kalliatakis, G. (2019). GET-AID: Visual Recognition of Human Rights Abuses via Global Emotional Traits. https://github.com/GKalliatakis/GET-AID/releases/tag/v0.1-alpha. Accessed: 14 May 2019.

[40] Kalliatakis, G., Ehsan, S., Fasli, M., and D McDonald-Maier, K. (2019a). DisplaceNet: Recognising Displaced People from Images by Exploiting Dominance Level. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[41] Kalliatakis, G., Ehsan, S., Fasli, M., Leonardis, A., Gall, J., and McDonald-Maier, K. D. (2017a). Detection of Human Rights Violations in Images: Can Convolutional Neural Networks Help? In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), (VISIGRAPP 2017)*, pages 289–296.

[42] Kalliatakis, G., Ehsan, S., Fasli, M., and McDonald-Maier, K. D. (2019b). Get-aid: Visual recognition of human rights abuses via global emotional traits. *arXiv preprint arXiv:1902.03817*.

[43] Kalliatakis, G., Ehsan, S., Leonardis, A., Fasli, M., and McDonald-Maier, K. D. (2019c). Exploring object-centric and scene-centric cnn features and their complementarity for human rights violations recognition in images. *IEEE Access*, 7:10045–10056.

[44] Kalliatakis, G., Ehsan, S., and McDonald-Maier, K. D. (2017b). A paradigm shift: Detecting human rights violations through web images. In *Proceedings of the Human Rights Practice in the Digital Age Workshop*.

[45] Kalliatakis, G., Stamatiadis, G., Ehsan, S., Leonardis, A., Gall, J., Sticlaru, A., and McDonald-Maier, K. D. (2017c). Evaluating deep convolutional neural networks for material classification. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), (VISIGRAPP 2017)*, pages 346–352.

[46] Kalliatakis, G., Sticlaru, A., Stamatiadis, G., Ehsan, S., Leonardis, A., Gall, J., and McDonald-Maier, K. D. (2018). Material Classification in the Wild: Do Synthesized Training Data Generalise Better than Real-world Training Data? In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), (VISIGRAPP 2018)*, pages 427–432.

[47] Karpathy, A. (2015). t-SNE visualization of CNN codes. Available from: https://cs.stanford.edu/people/karpathy/cnnembed/. Accessed: 25 April 2019.

[48] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732. IEEE Computer Society.

[49] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

[50] Koettl, C. (2016). Citizen media research and verification: an analytical framework for human rights practitioners. *Human Rights in the Digital Age: CGHR Practitioner Paper*.

[51] Kosti, R., Alvarez, J., Recasens, A., and Lapedriza, A. (2019). Context Based Emotion Recognition using EMOTIC Dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[52] Kosti, R., Alvarez, J. M., Recasens, A., and Lapedriza, A. (2017). Emotion recognition in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1960–1968. IEEE Computer Society.

[53] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

[54] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. Curran Associates, Inc.

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

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

[57] Leung, T. and Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International journal of computer vision*, 43(1):29–44.

[58] Li, Y., Yosinski, J., Clune, J., Lipson, H., and Hopcroft, J. E. (2015). Convergent learning: Do different neural networks learn the same representations? In *Proceedings of the Advances in Neural Information Processing Systems (NIPS) Workshop on Feature Extraction: Modern Questions and Challenges*, pages 196–212.

[59] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

[60] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2018). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[61] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer.

[62] Lipkin Stein, N., Dundes Renteln, A., Martinez Angela, M., Klupchak, A., Brown, S., Sherman, R., Gama, F., Graham, A., Wahlberg, K., Opalinski, A., Dyess, S., Stein, N., and Thompson, T. (2017). *Images and Human Rights: Local and Global Perspectives*. Cambridge Scholars Publishing, 1 edition.

[63] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer.

[64] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[65] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

[66] Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5188–5196. IEEE Computer Society.

[67] Marques, O., Barenholtz, E., and Charvillat, V. (2011). Context modeling in computer vision: techniques, implications, and applications. *Multimedia Tools and Applications*, 51(1):303–339.

[68] Marr, D. (1983). Vision: A computational investigation into the human representation and processing of visual information.

[69] Mehrabian, A. (1995). Framework for a comprehensive description and measurement of emotional states. *Genetic, social, and general psychology monographs*.

[70] Mehrabian, A. and Russell, J. A. (1974). *An approach to environmental psychology.* the MIT Press.

[71] Morgan, O. W., Sribanditmongkol, P., Perera, C., Sulasmi, Y., Van Alphen, D., and Sondorp, E. (2006). Mass fatality management following the South Asian tsunami disaster: case studies in Thailand, Indonesia, and Sri Lanka. *PLoS medicine*, 3(6):e195.

[72] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. *Deep Learning and Unsupervised Feature Learning Workshop, Neural Information Processing Systems Conference*.

[73] Olivas, E. S. (2009). *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques*. IGI Global.

[74] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1717–1724. IEEE Computer Society.

[75] Padania, S., Gregory, S., Alberdingk-Thijm, Y., and Nunez, B. (2011). Cameras everywhere: Current challenges and opportunities at the intersection of human rights, video and technology. *New York: Witness*.

[76] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[77] Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE Computer Society.

[78] Piracés, E., Land, M. K., and Aronson, J. D. (2018). The future of human rights technology: A practitioner's view. *New Technologies for Human Rights Law and Practice, edited by Molly K. Land and Jay D. Aronson*, pages 289–308.

[79] Quinn, J. A., Nyhan, M. M., Navarro, C., Coluccia, D., Bromley, L., and Luengo-Oroz, M. (2018). Humanitarian applications of machine learning with remote-sensing data: review and case study in refugee settlement mapping. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2128):20170363.

[80] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788. IEEE Computer Society.

[81] Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6517–6525. IEEE Computer Society.

[82] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99.

[83] Roth, K. (2019). Human Rights Watch (HRW). Available from: https://www.hrw.org/video-photos.

[84] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

[85] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626.

[86] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and Lecun, Y. (2014). OverFeat: Integrated recognition, localization and detection using convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

[87] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813. IEEE Computer Society.

[88] Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., and Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the European Conference on Computer Vision*, pages 510–526. Springer.

[89] Silverman, C. (2015). Lies, damn lies, and viral content: How news websites spread (and debunk) online rumors, unverified claims and misinformation. *Tow Center for Digital Journalism*, 168(4):134–140.

[90] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*.

[91] Snoek, C. G., Worring, M., and Smeulders, A. W. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM.

[92] Soleymani, M., Asghari-Esfeden, S., Fu, Y., and Pantic, M. (2015). Analysis of EEG signals and facial expressions for continuous emotion detection. *IEEE Transactions on Affective Computing*, 7(1):17–28.

[93] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

[94] Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 843–852.

[95] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9. IEEE Computer Society.

[96] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*.

[97] Tommasi, T. (2013). Learning to learn by exploiting prior knowledge. Technical report, EPFL.

[98] Tong, W., Yang, Y., Jiang, L., Yu, S.-I., Lan, Z., Ma, Z., Sze, W., Younessian, E., and Hauptmann, A. G. (2014). E-LAMP: integration of innovative ideas for multimedia event detection. *Machine vision and applications*, 25(1):5–15.

[99] Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171.

[100] United Nations High Commissioner for Refugees (UNHCR) (2017). Global Trends Forced Displacement in 2017. Technical report.

[101] UNOSAT-UNITAR (2014). Impact of the 2014 conflict in the gaza strip unosat satellite derived geospatial analysis. Available from: https://unosat.web.cern.ch/unosat/unitar/publications/UNOSAT_GAZA_REPORT_OCT2014_WEB.pdf. Accessed: 5 March 2019.

[102] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

[103] Walle, B., Eede, G., and Muhren, W. (2009). Humanitarian information management and systems. In *Mobile Response*, pages 12–21. Springer-Verlag.

[104] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. pages 3360–3367.

[105] Wang, L., Wang, Z., Du, W., and Qiao, Y. (2015). Object-scene convolutional neural networks for event recognition in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pages 30–35.

[106] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1):1–40.

[107] Weizman, E. (2017). *Forensic architecture: violence at the threshold of detectability*. MIT Press.

[108] Witmer, F. D. (2015). Remote sensing of violent conflict: Eyes from above. *International Journal of Remote Sensing*, 36(9):2326–2352.

[109] Wu, Z., Fu, Y., Jiang, Y.-G., and Sigal, L. (2016). Harnessing object and scene semantics for large-scale video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3112–3121. IEEE Computer Society.

[110] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE Computer Society.

[111] Y., U. (2015). Rightscon summit series. Available from: www.rightscon.org/about-and-contact/. Accessed: 16 January 2019.

[112] Yang, J., Yu, K., and Huang, T. (2010). Supervised translation-invariant sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3517–3524. IEEE Computer Society.

[113] Ye, G., Liu, D., Jhuo, I.-H., and Chang, S.-F. (2012). Robust late fusion with rank minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3021–3028. IEEE Computer Society.

[114] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328.

[115] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 818–833. Springer.

[116] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization.

[117] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2015). Object detectors emerge in Deep Scene CNNs. In *Proceedings of the International Conference on Learning Representations*.

[118] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929. IEEE Computer Society.

[119] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2018). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464.

[120] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495.

[121] Zhou, X., Yu, K., Zhang, T., and Huang, T. S. (2010). Image classification using super-vector coding of local image descriptors. In *European conference on computer vision*, pages 141–154. Springer.