Universitat Autònoma de Barcelona

Deep Metric Learning for re-identification, tracking and hierarchical novelty detection

Idoia Ruiz López

Universitat Autònoma
de Barcelona

Deep Metric Learning for re-identification, tracking and hierarchical novelty detection

A dissertation submitted by **Idoia Ruiz López** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor of Philosophy**.

Bellaterra, June 15, 2022

Director	**Dr. Joan Serrat Gual** Dept. Ciències de la computació & Centre de Visió per Computador Universitat Autònoma de Barcelona
Thesis committee	**Dr. Petia Radeva** Dept. de Matemàtiques i Informàtica Universitat de Barcelona **Dr. Ramon Badrich** Dept. Ciències de la computació & Centre de Visió per Computador Universitat Autònoma de Barcelona **Dr. Ferran Diego** Telefonica research

This document was typeset by the author using LaTeX 2$_\varepsilon$.

The research described in this book was carried out at the Centre de Visió per Computador, Universitat Autònoma de Barcelona.

A mi familia.

Acknowledgements

En primer lloc, vull agrair al meu supervisor Joan, per donar-me l'oportunitat de realitzar aquesta tesi i guiar-me durant tot el camí, ajudant-me sempre amb bona voluntat, paciència i dedicació en tot el que ha sigut possible. Les discussions i experiència durant aquests anys m'han fet millorar notablement i aprendre molt.

I am also thankful to the ADAS group for our interesting conversations and meetings. A Antonio, por ayudarnos a Joan y a mí siempre que ha sido necesario. And of course, thanks to the rest of the CVC people for their kindness during these years.

També estic agraïda als organitzadors del Màster en Visió per Computador. Iniciar aquests estudis va ser el punt d'inflexió que em va introduir en aquest camp, canviant significativament el meu futur. A la UAB, que ha permès amb el seu contracte que aquesta tesi es fes realitat.

I acknowledge the Mapillary team the warm welcome I received during my stay at Graz. They made me feel at home and I keep good memories of the months spent with them. I appreciate the opportunity of learning from such a knowledgeable team, which was a lot even in a short period of time.

Por último, agradezco el apoyo constante e incondicional de mi familia, que siempre me ha animado con todo lo que he hecho. Así como todos sus esfuerzos para que pudiese optar a todas las oportunidades que se me han presentado.

Abstract

Metric learning refers to the problem in machine learning of learning a distance or similarity measurement to compare data. In particular, deep metric learning involves learning a representation, also referred to as embedding, such that in the embedding space data samples can be compared based on the distance, directly providing a similarity measure. This step is necessary to perform several tasks in computer vision. It allows to perform the classification of images, regions or pixels, re-identification, out-of-distribution detection, object tracking in image sequences and any other task that requires computing a similarity score for their solution. This thesis addresses three specific problems that share this common requirement. The first one is person re-identification. Essentially, it is an image retrieval task that aims at finding instances of the same person according to a similarity measure. We first compare in terms of accuracy and efficiency, classical metric learning to basic deep learning based methods for this problem. In this context, we also study network distillation as a strategy to optimize the trade-off between accuracy and speed at inference time. The second problem we contribute to is novelty detection in image classification. It consists in detecting samples of novel classes, *i.e.* never seen during training. However, standard novelty detection does not provide any information about the novel samples besides they are unknown. Aiming at more informative outputs, we take advantage from the hierarchical taxonomies that are intrinsic to the classes. We propose a metric learning based approach that leverages the hierarchical relationships among classes during training, being able to predict the parent class for a novel sample in such hierarchical taxonomy. Our third contribution is in multi-object tracking and segmentation. This joint task comprises classification, detection, instance segmentation and tracking. Tracking can be formulated as a retrieval problem to be addressed with metric learning approaches. We tackle the existing difficulty in academic research that is the lack of annotated benchmarks for this task. To this matter, we introduce the problem of weakly supervised multi-object tracking and segmentation, facing the challenge of not having available ground truth for instance segmentation. We propose a synergistic training strategy that benefits from the knowledge of the supervised tasks that are being learnt simultaneously.

Key words: *metric learning, novelty detection, hierarchical classification, multi-object tracking, instance segmentation, person re-identification, autonomous driving, computer vision, machine learning*

Resumen

El aprendizaje de métricas se refiere al problema del aprendizaje automático de aprender una medida de distancia o similitud con el objetivo de comparar datos. En particular, el aprendizaje de métricas profundo implica aprender una representación de las imágenes tales que en su subespacio las muestras de datos se pueden comparar en función de la distancia, proporcionando directamente una medida de similitud. Este paso es necesario para realizar varias tareas en visión artificial. Permite realizar la clasificación de imágenes, regiones o píxeles, reidentificación, detección de muestras que no pertenecen a la distribución, seguimiento de objetos en secuencias de imágenes y cualquier otra tarea que requiera calcular una medida de similitud. Esta tesis aborda tres problemas específicos que comparten este requisito común. El primero es la reidentificación de personas. En esencia, es una tarea de recuperación de imágenes que tiene como objetivo encontrar instancias de la misma persona en base a una medida de similitud. Primero comparamos, en términos de precisión y eficiencia, el aprendizaje de métricas clásico contra métodos básicos de aprendizaje profundo para este problema. En este contexto, también estudiamos la destilación de redes como una estrategia para optimizar el intercambio entre precisión y velocidad de inferencia. El segundo problema al que contribuimos es la detección de novedades en la clasificación de imágenes. Consiste en detectar muestras de clases nuevas, es decir, nunca vistas durante el entrenamiento. Sin embargo, la detección de novedades estándar no proporciona ninguna información sobre las muestras desconocidas más allá de que lo son. Con el fin de obtener resultados más informativos, aprovechamos las taxonomías jerárquicas presentes de forma intrínseca en las clases. Nuestro enfoque basado en el aprendizaje de métricas aprovecha las relaciones jerárquicas entre las clases durante el entrenamiento, pudiendo predecir la clase padre en la jerarquía de una muestra desconocida. Nuestra tercera contribución es el seguimiento y la segmentación de múltiples objetos. Esta tarea conjunta comprende clasificación, detección, segmentación de instancias y seguimiento. El seguimiento se puede formular como un problema de recuperación que se abordará con aprendizaje de métricas. Abordamos una dificultad existente en la investigación académica, que es la falta de bases de datos anotados para esta tarea. Introducimos el problema del seguimiento y segmentación de múltiples objetos débilmente supervisado, enfrentándonos al desafío de no tener anotaciones disponibles para la segmentación de instancias. Proponemos una estrategia sinérgica de entrenamiento que se beneficia del conocimiento extraído de las tareas supervisadas que se están aprendiendo simultáneamente.

Palabras clave: *aprendizaje de métricas, detección de novedades, clasificación jerárquica, seguimiento de múltiples objetos, segmentación de instancias, reidentificación de personas, conducción autónoma, visión artificial, aprendizaje automático*

Resum

L'aprenentatge de mètriques es refereix al problema de l'aprenentatge automàtic d'aprendre una mesura de distància o similitud amb l'objectiu de comparar dades. En particular, l'aprenentatge de mètriques profund implica aprendre una representació de les imatges tals que al seu subespai les mostres de dades es poden comparar en funció de la distància, proporcionant directament una mesura de similitud. Aquest pas és necessari per a resoldre diverses tasques en visió artificial. Permet realitzar la classificació d'imatges, regions o píxels, reidentificació, detecció de mostres que no pertanyen a la distribució, seguiment d'objectes en seqüències d'imatges i qualsevol altra tasca que requereixi calcular una mesura de similitud. Aquesta tesi aborda tres problemes específics que comparteixen aquest requisit comú. El primer és la reidentificació de persones. En essència, és una tasca de recuperació d'imatges que té com a objectiu trobar instàncies de la mateixa persona basant-se en una mesura de similitud. Primer comparem, en termes de precisió i eficiència, l'aprenentatge de mètriques clàssic contra mètodes bàsics d'aprenentatge profund per a aquest problema. En aquest context, també estudiem la destil·lació de xarxes com una estratègia per a optimitzar l'intercanvi entre precisió i velocitat d'inferència. El segon problema al qual contribuïm és la detecció de novetats en la classificació d'imatges. Consisteix a detectar mostres de classes noves, és a dir, mai vistes durant l'entrenament. No obstant això, la detecció de novetats estàndard no proporciona cap informació sobre les mostres desconegudes més enllà que ho són. Amb la finalitat d'obtenir resultats més informatius, aprofitem les taxonomies jeràrquiques presents de manera natural en les classes. El nostre enfocament basat en l'aprenentatge de mètriques aprofita les relacions jeràrquiques entre les classes durant l'entrenament, podent predir la classe pare en la jerarquia d'una mostra desconeguda. La nostra tercera contribució és el seguiment i la segmentació de múltiples objectes. Aquesta tasca conjunta comprèn classificació, detecció, segmentació d'instàncies i seguiment. El seguiment es pot formular com un problema de recuperació que s'abordarà amb aprenentatge de mètriques. Abordem una dificultat existent en la recerca acadèmica, que és la falta de bases de dades anotades per a aquesta tasca. Introduïm el problema del seguiment i segmentació de múltiples objectes feblement supervisat, enfrontant-nos al desafiament de no tenir anotacions disponibles per a la segmentació d'instàncies. Proposem una estratègia sinèrgica d'entrenament que es beneficia del coneixement extret de les tasques supervisades que s'estan aprenent simultàniament.

Paraules clau: *aprenentatge de mètriques, detecció de novetats, classificació*

jeràrquica, seguiment de múltiples objectes, segmentació d'instàncies, reidentificació de persones, conducció autònoma, visió artificial, aprenentatge automàtic

Contents

List of Figures

List of Tables

1 Introduction

1.1 Metric Learning

Machine learning based technologies have rapidly advanced in the past years due to the increasing interest of the research community in this field. Among the tasks these techniques can be applied to, computer vision related problems are some of the current hot topics. They address real-world challenges such us medical diagnosis, improving surveillance systems, developing autonomous driving perception, and broadly, automate any image-based task consisting in predicting certain output from information that is extracted from images. Deep learning are the state-of-the-art techniques on these computer vision tasks, having obtained excellent performance on them.

An important objective in several tasks of machine learning is the ability to compare data. For instance, some classification algorithms such as k-nearest neighbors rely on distance comparison. Also, any image retrieval method depends on similarity comparison among data instances. This is achieved by learning similarity measures or distance metrics. *Metric learning* refers to the problem of learning a distance function to be used for a particular task that requires similarity or distance estimation [46, 92]. This component is common to several computer vision tasks, *e.g.* retrieval tasks (face verification, person re-identification, tracking, etc), classification, clustering, and essentially, any problem that needs to compare features of instances within an embedding. In this kind of tasks, images are mapped to an embedding of features, where they can be ordered or classified. Metric learning allows to perform this feature comparison meaningfully, consequently increasing the performance on the target task. Its objective is to provide a better measurement for the considered problem. Such measurement can be learnt through leveraging the supervised data that is already available to solve the task. This data might contain the exact distance to predict for a pair of images (as in a regression problem), or we could use instead weaker constraints, *e.g. one pair of images is more similar than another*.

Traditionally, metric learning algorithms have been formulated as learning a linear mapping of the data to a discriminant embedding space. Many initial works emulate a Mahalanobis distance formulation, thus learning a projection matrix

that defines the linear transformation. This embedding space is often of lower dimensionality but tries to preserve the relevant information of feature description that enable comparison of different instances. However, this formulation cannot capture non-linear transformations and deal with more complex data distributions effectively. Kernel approaches were first introduced to overcome this limitation. But the great potential of the forthcoming deep learning based approaches would focus the interest of research community on them, overshadowing previous non-linear methods.

After the emergence of deep learning, previous formulations were substituted due to the large improvement on performance introduced by these techniques. Their advantages lie in the abilities to easily learn non-linear mapping functions introduced by the activation layers, and to work on large amounts of data, that results into better generalization. Nowadays, deep metric learning [41] methods constitute the state of the art. The first seminal work that considered to join convolutional neural networks with metric learning was applied to face verification [22]. The authors proposed a siamese architecture, consisting in two parallel network branches with shared weights. Their loss assigns high similarity to images of the same identity and low similarity to those that correspond to different people, by employing similar and dissimilar examples. This work opened the line of research of the posterior popular contrastive approaches. These build embeddings based on similarity relationships among samples, *e.g.* similar and dissimilar samples are compared to guide the learning. The objective is a discriminative embedding space, where features of similar samples are pushed together while those of dissimilar samples are far away. This idea is adopted by the contrastive loss [32]. It employs pairs of samples to learn the embedding and adds a margin parameter to enforce separation of the features. Later, the (still currently) widely used triplet loss was proposed in [37]. Instead of comparing pairs, it employs triplets that satisfy that two of the samples are more similar among them than to the other. These constraints build an embedding that yields better performance. For both formulations, mining strategies of samples have been investigated to effectively improve the learnt embedding [69, 84, 88]. They aim to find difficult or challenging pairs or triplets, for a faster convergence that leads to better minima. Later on, more recent works have proposed diverse similar losses aiming at improving its performance on different tasks [19, 68, 76]. However, approaches based on this contrastive formulation have the weaknesses of highly relying on the selection of pairs/triplets for a proper convergence, and they do not always achieve a discriminant feature space where features of the same class are mapped together and far from the other samples, as discussed in [88].

Non-contrastive deep metric learning approaches were later proposed as an alternative to deal with these issues. These loss functions, differently from the

aforementioned methods, do not rely on sample selection. The work of [62] in the field of face recognition, started a new family of methods. The authors proposed a modification of the softmax loss that includes a new angular margin hyperparameter to push the class decision boundaries, then increasing the inter-class variance. The softmax loss, in this context, refers to a cross entropy loss preceded by a softmax activation and a fully connected layer. In its original formulation, it already learns an embedding space that allows to distinguish different classes according to their distance. However, this space is not optimized to perform a distance-based classification or retrieval. The latter issue is what this family of methods address, aiming at pushing the discriminative power of the embeddings to later perform distance-based retrieval. SphereFace [61] applied normalization to the weights of the last fully-connected layer on the L-Softmax loss, so that they lie on a hyphersphere. This is specially useful in face verification because this retrieval task uses the cosine distance at test time. Similar approaches [24, 99, 100, 115, 119] proposed further improvements pursuing the same goal of increasing the discriminative power of the embeddings.

In the next sections, we introduce the different topics covered in this dissertation. They approach different problems and applications in computer vision, but have in common that include a deep metric learning component in their solution. From classical to contrastive or softmax based formulations, we study some of the aforementioned variants and apply them on challenging problems of computer vision, showing the potential and broad applicability of metric learning.

1.2 Person re-identification

Finding a person across a camera network plays an important role in video surveillance. Person re-identification [122] is the problem of, given a person of interest, retrieving all the available observations of that person at different camera views and times. This relevant problem with direct application to real-life, is essentially a retrieval task with a metric learning core. Its objective is to compare a person of interest against a gallery of many other samples. To perform such comparison, we clearly need a metric learning component.

As a starting point, in this thesis we aim to study an application of classical metric learning. By classical methods we refer to those developed before the *deep learning era*, *i.e.* they do not employ Convolutional Neural Network (CNN) architectures for visual description, as in the current state-of-the-art. Classical person re-identification approaches are usually composed by hand-crafted feature descriptors and appropriate distance functions that take into account the problem characteristics. Although Euclidean distance can be used for feature comparison,

incorporating a supervised metric learning component leads to more discriminative embeddings that provide a better measurement for this task. This results into greater performance, as we will see in chapter 2.

A second objective is to compare classical metric learning to a naive deep learning based approach in its simplest form, that is using the Euclidean distance on cross-entropy trained features from a CNN. As previously discussed, cross-entropy training of a fully connected layer followed by a standard softmax function, already leads to embeddings that can be distinguished based on Euclidean or cosine distance, although not being optimal for this objective. We consider this simple baseline to be compared against a classical person re-identification method. In particular, we consider LOMO and XQDA [55], which are specifically designed for this problem. In chapter 2, we investigate if the superior feature representations from CNNs, compared to hand-crafted ones, can compensate the lack of a dedicated supervised metric learning component.

From the perspective of a real-world person re-identification application, in order to guarantee an optimal time response, it is crucial to find the balance between accuracy and speed. Besides providing excellent performance, the computational time the methods takes to fulfill the task is another variable to optimize. A final objective we tackle is to analyze this trade-off on both hand-crafted and deep learning based techniques. Additionally, we propose network distillation as a learning strategy to reduce the computational cost of the deep learning approach at test time. We aim to show how distillation may help reducing the computational cost at inference time and its effect on the accuracy. The content of chapter 2 is based on our publication in [81].

1.3 Hierarchical novelty detection

Image classification has been thoroughly studied in the literature, resulting into impressive performance achieved by deep learning methods in this task. However, the problem of classifying what is *unknown* remains unsolved. Broadly, novelty or anomaly detection, also named open set recognition, deals with the problem of how to endow a classifier with the ability of knowing what it does not know. More precisely, *novelty detection* consists in the detection of unknown classes. Its objective is to detect samples of novel classes, never seen during training, while classifying those that belong to known classes. Recent works have made significant progress in novelty detection. It has been commonly addressed by probability based methods that employ the pre-softmax output of CNNs to perform the novel/known decision. However, we believe metric learning is an alternative that could generalize better on new unseen data. Contrastive approaches do not depend on the classes

themselves but on learning similarity/dissimilarity among samples. Therefore, they are suitable to be extended to new unknown classes. This motivated us to consider metric learning for novelty detection.

Figure 1.1: 3D projection of an embedding learned with a metric learning based novelty detection method [66]. Top right: Embedding of known classes. Top left: Embeddings of both known (blue) and novel samples (orange). Bottom: Same applied on a traffic signs dataset, showing thumbnails of the samples shaded in blue for the known samples and orange for the novel ones.

In prior work [66], we studied the feasibility of a metric learning based approach against cross-entropy based techniques, on a classification framework. Figure 1.1 shows an example of the learned embedding. The image shows a projection to three

dimensions by applying PCA. Known samples of the same class are pushed together in the embedding space, while novel samples lie far from the known class clusters. However, the only information standard novelty detection provides about novel samples is that they are *unknown*. Continuing this line of research, we set a new greater goal: to extend a metric learning based approach for novelty detection to a hierarchical setting. We aim at performing classification and novelty detection under hierarchical taxonomies of classes, which is a scarcely explored case.

In this dissertation, we leverage hierarchical taxonomies of classes to provide informative outputs for samples of novel classes, in particular, we predict the closest class in the taxonomy, *i.e.* its parent class. In chapter 3, we address this problem, known as hierarchical novelty detection [47], by proposing a novel loss, namely Hierarchical Cosine Loss (HCL). Inspired by previous face recognition literature [99, 100], HCL is designed to learn class *prototypes* along with an embedding of discriminative features consistent with the taxonomy.

Additionally, we aim to apply the developed methods in this thesis to the specific application of autonomous driving. Autonomous driving is one of the most significant technological challenges currently. Computer vision techniques have greatly contributed to its developement in recent years. Among its related problems, it comprises object detection, traffic sign recognition, road segmentation, tracking, etc. In chapter 3, we address the task of traffic sign recognition. Due to the intrinsic hierarchical nature of the class taxonomy of traffic signs, we specifically apply our developed methods to predict the parent class semantics for novel types of traffic signs. The content of chapter 3 is based on our publication in [82].

1.4 Weakly Supervised Multi-Object Tracking and Segmentation

Besides traffic sign recognition, we aim to explore other autonomous driving related problems. In the context of environment perception, object detection, instance segmentation or tracking are some of the relevant tasks to be solved. Multi-object tracking and instance segmentation (MOTS) [97] consists in detecting, classifying, tracking and predicting pixel-wise masks for the object instances present along a street-level video sequence. It therefore comprises several tasks that are solved jointly, *i.e.* object detection, classification, instance segmentation and tracking.

Among them, tracking can be either addressed as a tracking-by-detection approach or as a detection free tracking problem. The latter requires manual initialization of the object locations in the first frame, therefore not being able to deal with new objects entering the scene [10, 50, 51]. Differently, tracking-by-detection

Figure 1.2: Multi-object tracking and instance segmentation output: detection, classification, tracking and instance segmentation of the object instances present along a street-level video sequence.

[11, 89, 97, 104] relies on a previous object detector and a posterior linking strategy of the candidates. Both formulations actually correspond to a retrieval task that can be solved by a metric learning based method [10, 50, 51, 89, 97, 104]. Finding correspondences is essentially a retrieval task, that might be restricted to certain sets of candidates with additional information of the most likely ones.

Due to the expensive cost of the annotation procedure, one of the limitations of MOTS is the lack of existing labeled data, necessary for training effectively a model that is able to solve the joint task. Joint tracking and segmentation requires annotations for both problems simultaneously. For this reason, the literature is still scarce on this topic. To overcome this limitation, we initially proposed an automatic annotation procedure for MOTS benchmarks in [74]. In that work we investigated an unsupervised approach for the tracking task. In this dissertation, we instead intend to explore a weakly supervised setting of the instance segmentation task.

In chapter 4, we introduce the problem of weakly supervised Multi-Object Tracking and Segmentation [80], *i.e.* joint weakly supervised instance segmentation and multi-object tracking. In this setting we do not provide any kind of mask annotation, which is the most expensive kind. To this end, we employ a popular

strategy in the weakly-supervised instance segmentation field, that consists in using the activation maps from the classification task [86]. We extract weak foreground localization information, provided by Grad-CAM heatmaps, to generate a partial ground truth to learn from. Additionally, RGB image level information is employed to refine the mask prediction at the edges of the objects. We benefit from our multi-task problem, so that the supervised tasks, *i.e.* classification, object detection and tracking, guide the learning of unsupervised instance segmentation. The employed model is MOTSNet [74], that similarly to [97], is Mask-RCNN based with an additional tracking head that is trained by using metric learning. The content of chapter 4 is based on our publication in [80].

1.5 Objectives and Scope

The aim of this PhD dissertation is to explore the capabilities of deep metric learning and further apply it to diverse problems in computer vision. We study different related tasks, such as retrieval, classification, novelty detection or tracking.

The first problem we address in chapter 2 is person re-identification. We aim at answering the following research questions:

- How important is the metric learning component in this task?
- In terms of efficency and accuracy, how do hand-crafted approaches (*i.e.* classical metric learning) compare to basic deep learning based ones?
- Can we improve this trade-off for deep learning based methods by using network distillation?

The next chapters address the main application covered in this dissertation, *i.e.* autonomous driving. As previously discussed, in this thesis we consider the problems of traffic sign recognition and scene perception via multi-object tracking and segmentation. For traffic sign recognition, we aim to leverage the natural hierarchical taxonomy of classes which traffic signs are organized by. For this purpose, we ask the following questions:

- How do hierarchical taxonomies of classes can be exploited for informative novelty detection?
- Is it possible to perform hierarchical novelty detection by a metric learning based approach? How does it compare to probability based methods?

Regarding multi-object tracking and segmentation (MOTS), we aim at finding a solution for the lack of labeled data. In particular, we study this problem under

a weakly supervised setting where there is no available ground truth for instance segmentation. In this context, our research questions are:

- Can we benefit from the joint tasks that are simultaneously solved in MOTS to provide information for the weakly supervised instance segmentation task? How?
- What components might help to this objective and to what extent?

1.6 Outline

This PhD thesis is structured as follows. On each chapter we address a different problem that includes a metric learning component for its solution. They are self-contained and include: a corresponding concrete introduction to the problem, an analysis of the related works, description of the method, experimental evaluation and individual conclusions.

In chapter 2, we address the problem of person re-identification. We first investigate classical metric learning in contrast to most basic deep learning methods. Moreover, we propose network distillation to improve the accuracy/speed trade-off of the pipeline. In chapter 3, we deal with hierarchical novelty detection. We propose a metric learning based alternative, in contrast to current probability based state-of-the-art approaches. We additionally present a specific application to traffic sign recognition. Chapter 4 introduces the problem of weakly supervised Multi-Object Tracking and Segmentation [80], *i.e.* joint weakly supervised instance segmentation and multi-object tracking. Our synergistic training strategy takes advantage of multi-task learning to solve unsupervised instance segmentation. To conclude, chapter 5 collects the general conclusions drawn from this dissertation and indicates possible directions for future work. It finally includes the list of publications made throughout the development of this thesis.

2 Person Re-identification

2.1 Introduction

Person re-identification refers to the problem of identifying a person of interest across a camera network [70, 123]. This task is specially important in surveillance applications, since nowadays the security systems in public areas such as airports, train stations or crowded city areas, highly rely on video monitoring and are continuously improving to ensure the population's welfare. In big cities, there are extensive networks of cameras in the most sensitive locations. Identifying an individual requires finding it among all the instances that are present on the collection of images captured by the cameras. These images show usually complex crowded scenes, which further increases the computational complexity of the problem. Therefore, the automation of this task involving large-scale data becomes essential, as otherwise it would be a laborious task to be performed by humans.

The aim of person re-identification is to find a person of interest, also referred as *query*, across a *gallery* of images. The difficulty of this problem lies in variations in the point of view, person pose, light conditions and occlusions that affect the images. This kind of variability is illustrated on Figure 2.4, that shows examples of gallery images.

A full person re-identification system, including the previous person detection stage, is depicted in Figure 2.1. Within the person re-identification module, a *query* image of a person of interest is compared against the *gallery*, to retrieve the images that correspond to the same identity. The system first extracts a feature representation that describes every image, either by using a hand-crafted descriptor or a deep neural network. Usually the features of the *gallery* are previously computed offline and stored, so that at test time only the features for the query image are computed. These can be compared with the features of the *gallery* by employing a similarity metric, thus obtaining a ranked list of the most similar images in the gallery to the person of interest [129], according to the degree of similarity.

In real-life scenarios, in order to have a feasible application that is able to work with large-scale datasets in an efficient and effective way, we have to address the problem of optimizing the computational cost of the system at test time, without decreasing drastically its accuracy. For that purpose, we consider both classical

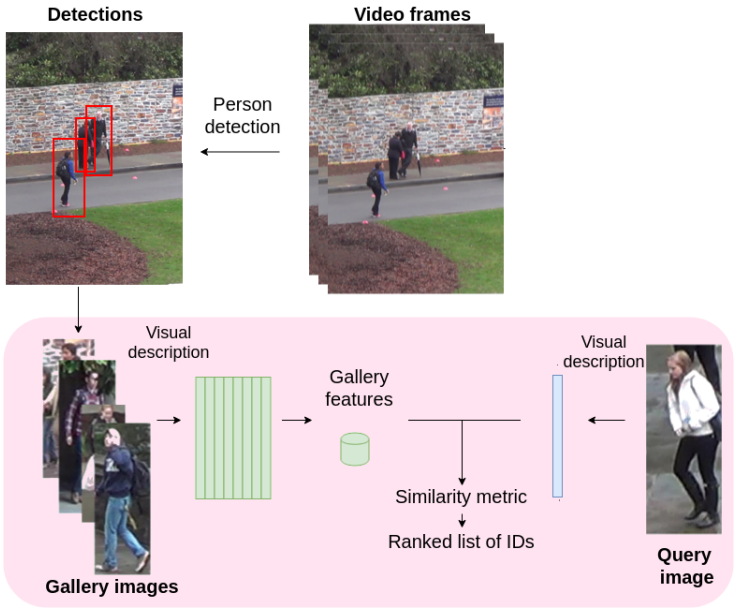


Figure 2.1: Pipeline of an end-to-end person re-identification system. The pink shaded region delimits the person re-identification module.

and deep learning based person re-identification methods. Although deep learning based techniques outperform significantly hand-crafted methods in terms of accuracy, their drawback is that they require dedicated hardware, *i.e.* GPUs, and big amounts of data for training, which takes usually long periods of time, *i.e.* weeks, in order to be effective.

To make deep learning approaches computationally efficient, several works use model compression [5, 16]. The idea behind model compression is to discard *non-informative* weights in the deep networks and perform a fine-tuning to further improve the performance. Although these methods make the architecture more efficient in terms of computational complexity, they also result in a drop of the accuracy on the compressed models. This drop is specially prominent when the dataset is large or the number of classes is higher, which is often the case in the person re-identification problem. In contrast, network distillation works have shown that the smaller or compressed model trained with the support of a much bigger/deeper network is able to achieve very similar accuracy as the deeper network

but having a much lower complexity [5, 78, 114]. Motivated by this good performance, in this chapter we explore network distillation in the context of efficient person-re-identification.

In this chapter, we provide an analysis of the trade-off between accuracy and computational cost at test time in the person re-identification problem, considering the most suitable configuration for the conditions of a real-world application. We carry out such trade-off analysis on two challenging large-scale person re-identification benchmarks, *i.e.* Market-1501 [121] and DukeMTMC-reID [77]. Moreover, we introduce and analyze network distillation [36] for optimizing this trade-off for the deep learning approach. For this purpose, we use a ResNet-50 [33] model, acting as the teacher, to transfer the knowledge to a lighter model, in our case MobileNet [38], acting as the student.

This chapter is structured as follows. In Section 2.2, we review the literature related with person re-identification and distillation. In Section 2.3 we describe briefly the methods employed in our experiments, then reviewing the distillation approach in Section 2.4. The experimental results are reported in Section 2.5. Finally, in Section 2.7, we present our conclusions and provide some guidelines for future work.

2.2 Related Work

2.2.1 Person re-identification

Classical methods for person re-identification considered it as two independent subsequent problems, *i.e.* feature representation and metric learning. Historically, for the first task of visual description, popular frameworks such as Bag of Words [126] or Fisher Vectors [72] were initially used to encode the local features. Later, other hand-crafted features were introduced. For instance, the LOMO [55] descriptor became popularized for person re-identification [70, 96, 129]. In the exhaustive comparison performed by [40], LOMO is the second hand-crafted feature descriptor that performs best across several datasets. The GOG [67] features are superior in terms of accuracy at the cost of a higher computational cost, as it requires modeling each subarea in which the image is divided, by a set of Gaussian distributions. Indeed, in [67], LOMO features are extracted in 0.016 seconds/image, while GOG features are extracted in 1.34 second/image.

The second task, metric learning, consists in learning a distance function that maps from the feature space to a new space in which the vectors of features that correspond to the same identity are close, while those that correspond to different identities are farther away, being the distance a measure of the similarity. Specifi-

cally on person re-identification, this mapping function after being learned is used to measure the similarity between the features of the person of interest and the gallery images. One of the most popular classical metrics is KISSME [42], that uses the Mahalanobis distance. Later, XQDA[55] was introduced as an extension of KISSME to cross-view metric learning, instead doing the mapping function from the feature space to a lower dimensionality space, in which the similarity metric is computed. More recently, [3] proposed a novel metric learning method that address the small sample size problem, which is due to the high dimensionality of the features on person re-identification. According to this metric, the samples of distinct classes are separated with maximum margin while keeping the samples of same class collapsed to a single point, intended to maximize the separability in terms of Fisher criterion.

Nowadays, deep learning based methods are outperforming hand-crafted techniques by a large margin. First approaches [6, 109, 122] used deep learning only to compute better image representations, then employing the similarity metric as usual. Considering each identity as a different class, the features are extracted from a classification Convolutional Neural Network (CNN), that is trained on the target dataset. The features, that we refer to as *deep features*, are the logits, *i.e.* the output of the network before the classification layer. A more complex framework was proposed in [52], where using a multi-scale context-aware network, they compute features that contain both global and local part-based information.

In a different line of work, siamese models were proposed to learn jointly the representations along with computing the similarity between the inputs, that are image pairs. The similarity measure provided by the output of the network, determines whether the input images correspond to the same identity or not. This architecture was first introduced by [13] for signature verification, where the features for two signature images were extracted and compared by computing the cosine of the angle between the two feature vectors as a measure of the similarity. Similarly, in person re-identification, siamese networks take as an input two person images. This original approach is followed in [113]. Other architectures such as [54] or [2] used the softmax layer to provide a binary output. Later, based also on a siamese framework, the authors of [124] proposed an architecture with an enhanced attention mechanism, in order to increase the robustness for cross-view matching. Closely related to siamese networks, triplet networks, which were introduced in [84] for face recognition, take triplets of images as inputs, corresponding only two of them to the same person [20, 117, 120]. Following a similar reasoning, a quadruplet loss was then proposed in [19].

Recent approaches aim at increasing the robustness of person re-identification systems. Some address the problem of domain adaptation, *i.e.* applying to an unseen dataset a model is trained on a set of source domains without any model

updating [60, 91]. To this end, image synthesis [25, 130] or domain alignment [59, 101, 102] are used. Other works instead propose generative approaches for data augmentation. In [75] the synthesized images help learning view-point invariant features by normalizing across a set of generated enhanced pose variations, while in [127] they compose high-quality cross-identities images.

2.2.2 Network Distillation

Network distillation approaches appeared as a computational effective solution to transfer the knowledge from a large, complex neural network (referred to as *teacher network*) to a more compact one (*student network*), with a significant lower number of parameters. This idea was originally proposed in [36]. On their approach, the student network was penalized based on a softened version of the teacher network's output. The student was trained to predict the output of the teacher, as well as the true classification labels. In [78], they proposed an idea to train a student network which is deeper and thinner than the teacher network. They do not only use the outputs, but also the intermediate representations learned by the teacher as hints to improve the training process and final performance of the student. A different approach was proposed in [65], where the knowledge to be transferred from the teacher to the student is obtained from the neurons in the top hidden layer, which preserve as much information as the softened label probabilities, but being more compact.

Network distillation approaches have also been applied recently to the person re-identification problem. In [116], the authors propose using a pair of students to learn collaboratively and teach each other throughout the training process. Each student is trained with two losses: a conventional supervised learning loss, and a mimicry loss that aligns each student's class posterior with the class probabilities of other students. This way, each student learns better in such peer-teaching scenario than when learning alone. In [30], feature distillation is used to learn identity-related and pose-unrelated representations. They adopt a siamese architecture, consisting each branch of an image encoder/decoder pair, for feature learning with multiple novel discriminators on human poses and identities. The recent work in [105] resembles ours in some aspects, although their scope is semi-supervised and unsupervised person re-identification, in contrast to our fully-supervised formulation. Similarly to us, they consider lightweight models to reduce testing computation as well as network distillation as an strategy of knowledge transfer. However, their distillation approach is not probability based, but similarity based. They propose the Log-Euclidean Similarity Distillation Loss that mimics the pairwise similarity of the teacher instead of using soft labels as we do. They explore a multiple teacher-single student setting and propose an adaptive knowledge aggregator to weight the

contributions of the teachers.

2.3 On Metric Learning in Person Re-identification

2.3.1 Classical methods

Classical methods approach person re-identification as two independent problems, that are feature representation and metric learning (see Section 2.2.1 for a description of some of them). Hand-crafted feature descriptors are designed by taking into account the nature of the problem and its data.

In our experiments, we consider the LOMO feature descriptor to work jointly with the XQDA metric learning algorithms [55], because they aim at being effective and computationally efficient. As discussed in Section 2.2.1, LOMO presents the best trade-off between accuracy and computational cost among all the methods considered in the exhaustive analysis performed in [40].

LOMO features are claimed to be robust against view changes and illumination variations. The method is based on extracting at different scales and locations of the image, features that encode color information via HSV histograms plus texture description computed by the SILTP [56] descriptor. After concatenating features from different scales, the resulting feature vectors have a dimensionality of 26960.

To deal with this high dimensionality, [55] also propose **XQDA** as the metric learning algorithm. It is basically an extension of KISSME [42] to cross-view data. These approaches involve simultaneously learning a discriminant subspace along with a metric. XQDA consists in a mapping to a lower dimensional space in which a quadratic discriminant analysis is performed using cross-view data.

In this analysis, two classes of variations are considered: the intrapersonal variations Ω_I, *i.e.* the variations between samples that correspond to the same identity, and the extrapersonal variations Ω_E, *i.e.* the variations between samples that correspond to different identities. These variations are simply computed as the difference between features of different samples. The objective is to learn a mapping from the feature space of high dimensionality d to a subspace $W \in \mathbb{R}^{d \times r}$ of lower dimensionality r. The projection matrix $W = (\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_r})$ is learned by maximizing $J(\mathbf{w}) = \frac{\sigma_E(\mathbf{w})}{\sigma_I(\mathbf{w})}$, where $\sigma_I(\mathbf{w}) = \mathbf{w}^T \Sigma_I \mathbf{w}$ and $\sigma_E(\mathbf{w}) = \mathbf{w}^T \Sigma_E \mathbf{w}$, being Σ_I and Σ_E the covariance matrices of Ω_I and Ω_E, respectively. The problem of optimizing $J(\mathbf{w})$ is solved by computing its eigenvalue decomposition. Then, the solution is given by the r largest eigenvalues of $J(\mathbf{w})$. Note that the selection of r determines the dimensionality of the subspace. The authors propose to only take the eigenvalues that are greater than 1, since smaller values correspond to the cases where $\sigma_E < \sigma_I$, thus not providing discriminant information. Finally, the computed distance d_W in

the new XQDA space is defined as (2.1),

$$d_W(\mathbf{x},\mathbf{z}) = (\mathbf{x}-\mathbf{z})^T W(\Sigma'^{-1}_I - \Sigma'^{-1}_E)W^T(\mathbf{x}-\mathbf{z}), \tag{2.1}$$

where $\mathbf{x}$ and $\mathbf{z}$ are features of samples that belong to different views and $\Sigma'_E = W^T\Sigma_E W$, $\Sigma'_I = W^T\Sigma_I W$.

2.3.2 Deep features

Deep learning based approaches employ the feature representations extracted from a CNN, as described in more detail in Section 2.2.1. The CNN is trained by considering the different identities as disjoint classes and benefits from large-scale datasets, which allow better generalization. The *deep features* correspond to the output of the last layer before the softmax. To compare different gallery images, we normalize the features then computing their Euclidean distance. This measurement provides a similarity metric which ranking is consistent with the cosine similarity. Although not being optimal, it enables the comparison among different instances providing a baseline for deep learning based approaches, as shown in previous works [109, 122]. Its good performance is consequence of the quality of the features rather than a metric learning algorithm, differently from classical approaches.

2.4 Reviewing Distillation

Besides improving the performance of the person re-identification pipeline in terms of computational cost at test time, we also aim at maximizing the performance of a small network to be as accurate as possible. As discussed in [36], the simplest way to transfer the knowledge is to use the output of the teacher network as soft targets for the student network, additionally to the hard targets provided by the ground truth. However, when the soft targets have high entropy, they provide more information to learn from. A network that is very confident about its prediction will generate a probability distribution similar to a Dirac delta function, in which the correct class has a very high probability while the rest of classes are predicted with almost zero probability. This probability distribution has very low entropy and consequently provides less information than a less confident network, which would assign higher probabilities to the incorrect classes, as shown graphically in Figure 2.2. The intuition behind high entropy distributions help the distillation, is that by learning from the probabilities assigned to incorrect classes, the student network is learning how the teacher model generalizes.

The objective is therefore to increase the entropy of the probability distribution generated by the teacher model, *i.e.* the output of the softmax layer, to leverage the

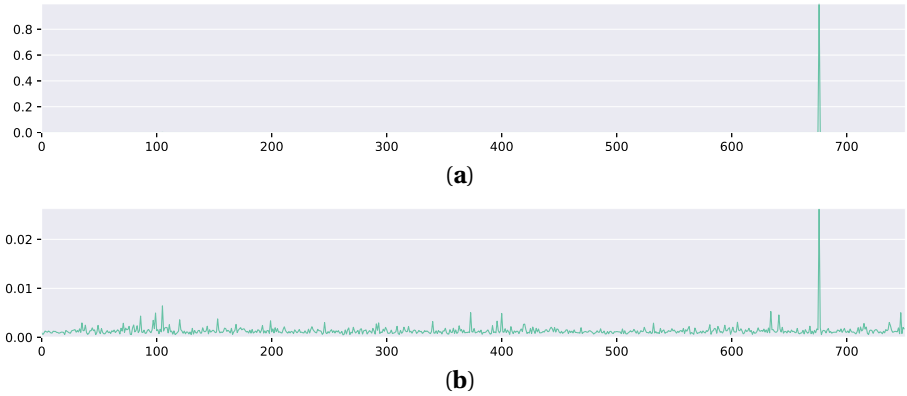


Figure 2.2: Example of (**a**) low and (**b**) high entropy probability distributions that are generated by the softmax layer of the teacher network for the Market-1501 dataset (751 classes). When the network has high confidence about the predictions, as in case (**a**), it provides hard targets with low entropy and therefore less information than if the network generates a probability distribution similar to the case (**b**), with the differences between the predicted probabilities of the incorrect classes being enhanced. This second case provides more information that is helpful for the distillation process.

information the student network learns from. In order to maximize the entropy, the authors, by drawing an analogy with statistical physics, propose to increase the *temperature* of this distribution. The *logits* z_i, that are the inputs of the softmax layer, are converted to probabilities p_i by the softmax function as follows,

$$p_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}, \tag{2.2}$$

where T is the temperature, that is a selected constant value in the distillation case, and it is equal to 1 when there is no distillation.

The knowledge transfer is performed via the loss function of the student model L_s. The loss function for the k-th training example L_{s_k} is the weighted sum of two terms and is defined as,

$$L_{s_k} = \underbrace{H\big(p_t(T=T_0), p_s(T=T_0)\big)}_{\text{Distillation term}} + \lambda \underbrace{H\big(y_k, p_s(T=1)\big)}_{\text{Cross-entropy loss}}, \tag{2.3}$$

where $H(p,q)$ denotes the cross-entropy between two probability distributions p and q. The first term is the cross-entropy between the soft targets extracted from the teacher $p_t(T = T_0)$ and the softened probability distribution of the student $p_s(T = T_0)$. $p_t(T = T_0)$ is the softened probability distribution of the teacher, that is obtained by applying the softmax function (2.2) to the logits of the teacher divided by a temperature T_0. For $p_s(T = T_0)$ we use the same T_0 value. The second term of the loss is the cross-entropy between the hard targets y_k, that is the ground truth distribution for the k-th sample, which has a value equal to 1 assigned to the correct class and 0 to the rest of them, and the probability distribution of the student ($p_s(T = 1)$), that is the output of the softmax using a $T = 1$. This second term is the standard cross-entropy loss function, which minimizes the cross-entropy between the prediction of the network and the ground truth. These two terms are balanced by a regularization parameter λ.

A graphical summary of the process is shown in Figure 2.3. In the current framework of person re-identification, once the student network is trained via distillation, it is used to extract the features of the images at test time, to then measure their similarity using the Euclidean distance.

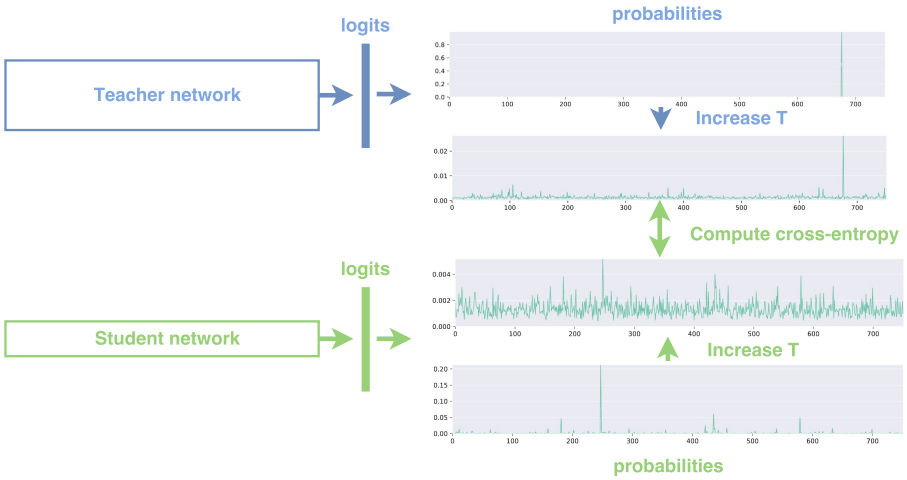


Figure 2.3: Distillation process. The cross-entropy between the softened distributions generated by the teacher and the student networks is computed in order to minimize it additionally to the cross-entropy with the ground truth.

2.5 Experiments

2.5.1 Datasets

In a real-world application, there are often several cameras that capture images of people from different points of view in different illumination conditions and even with occlusions. Market-1501 [121] or DukeMTMC-reID [77] have these characteristics that match a real-life scenerario, providing images taken from 6 cameras in the case of Market-1501 and 8 in the case of DukeMTMC-reID, that are captured in outdoor public areas. They are also two of the largest-scale public datasets for person re-identification. Sample images are shown in Figure 2.4.

(**a**)

(**b**)

Figure 2.4: Subset of gallery images that correspond to 2 identities from the (**a**) DukeMTMC-reID and (**b**) Market-1501 datasets. Each identity can appear in different cameras and may present different points of view, pose, and illumination conditions.

Market-1501 provides an average of 14.8 cross-camera ground truths for each query, containing in total 32,668 bounding boxes of 1,501 identities, from which 12,936 bounding boxes with 751 identities belong to the training set. The mean of images per identity is 17.2. All the bounding boxes are of size 128x64.

The DukeMTMC-reID benchmark is an extension of the DukeMTMC tracking dataset. The bounding boxes are then extracted from the full frames provided by the original dataset, resulting into person images of different sizes. It contains 36,441 bounding boxes that belong to 1,404 identities plus 408 distractor identities that only appear in a single camera. Among them, 16,522 bounding boxes with 702 identities are used for the training set. The mean number of images per identity

is 20, with a maximum of 426 images for the identity with the largest amount of images.

2.5.2 Evaluation

In a re-identification task, the *query* is compared to all the *gallery*, computing a similarity metric that is used to rank the *gallery* images sorted by similarity. The rank-1 accuracy gives the probability of getting a true match from the gallery in the first position of the ranking. Similarly, the rank-5 accuracy evaluates if we find a true match in the five first positions of the ranking. As the person of interest may appear many times in the gallery, we however need an evaluation metric that also considers finding all the true matches that exist in the gallery, evaluating also the recall. The mean average precision (mAP) is suitable for datasets in which an identity appears more than once in the gallery, such as Market-1501 and DukeMTMC-reID.

We additionally report the computational cost at test time of the algorithms proposed, by providing the time that feature extraction takes per image of a single individual. We extract the features for all the *gallery* and compute the average time per image. As the computational cost metric, we report the number of images the system extracts the features from in a second, for the different considered architectures. Then, the computational cost for the metric learning step is reported separately.

2.5.3 Implementation details

To analyze the trade-off between accuracy and computational cost at test time, we compare both classical and deep learning based approaches. In a real world application, both of them can be considered depending on the scenario and available resources.

Hand-crafted features

To evaluate the LOMO features independently to XQDA, we compare the Euclidean distance, KISSME [42] and XQDA as similarity metrics. Note that PCA is commonly applied previously to KISSME in order to reduce the dimensionality of the LOMO features, in our case from 26960 to 200. XQDA instead allows to select the dimensionality of its subspace, which enables to measure the performance of LOMO + XQDA depending on the XQDA dimensionality. The maximum value that we consider is the highest one with eigenvalues greater than 1. Following this criteria, we get a maximum dimensionality of 76 for the features extracted from the Market-1501 dataset. Therefore, we consider values of the XQDA dimensionality from 25 to 75. Finally, to evaluate the computational cost, we measure the inference time of the

method, running these experiments on a laptop with a CPU Intel Core i5-6300U CPU @ 2.40GHz.

Deep features

We take as a baseline the approach employed in [122] for the Market-1501 dataset, using the ResNet-50 [33] model. As ResNet-50 might be too large for the datasets we consider, we also explore other smaller networks that can be more efficient and still perform well. In particular, we consider MobileNets [38].

MobileNets are introduced as efficient light weight models suitable for mobile applications. The MobileNets architecture can be adapted to particular requirements of the system. In order to decide the network size, two parameters are introduced to control its latency and accuracy: the width multiplier $\alpha \in (0,1]$ and the resolution multiplier $\rho \in (0,1]$. The width multiplier can make the model thinner, by multiplying the number of input and output channels on each layer by α. ρ is implicitly selected when determining the input size of the network and can take the values of 224, 192, 160 and 128.

Our deep learning based methods are implemented using the TensorFlow library. The training and validation splits used for deep features are the ones provided on the original baselines. For Market-1501, [122] use a validation split of 1,294 images leaving 11,642 for training. The baseline for DukeMTMC-reID [128] uses the entire set of training images. Finally, to evaluate the computational cost, we measure the inference time, running the experiments on a NVIDIA GTX1070 GPU.

ResNet-50. The ResNet-50 network is fine-tuned from the weights pre-trained on ImageNet, considering the person identities as classes. The deep features are then extracted from the last layer before the softmax, which in the ResNet-50 architecture, corresponds to the output of the average pooling layer.

It is worth to mention that the high number of classes in the datasets (751 and 702 identities for the training splits of Market-1501 and DukeMTMC-reID respectively), with few images per class (a mean of 20 for DukeMTMC-reID and 17.2 for Market-1501), is a drawback to train the network since deep neural networks need a big enough amount of data to converge properly.

To train ResNet-50, we resize the input images to 224×224 and use horizontal flip for data augmentation. Using Stochastic Gradient Descent (SGD), we initially set the learning rate to 0.001 with a decay of 0.1 every 20000 steps. Using a batch size of 16 and momentum of 0.9, we train the network for 21 epochs (15000 iterations) for the Market-1501 dataset. For DukeMTMC-reID, the learning rate is initially set to 0.01 and we use a batch size of 32, training it for 29 epoch (15000 iterations).

MobileNets. We choose an input size of 128 because of the size of the images of the datasets we use. Market-1501 images have a fixed size of 128×64 while the size

of DukeMTMC-reID images varies. All the images are resized to 128×128, applying horizontal flip for data augmentation. We consider the width multiplier values of $\alpha = 0.25, 0.5, 0.75, 1.0$, which are those with available ImageNet pre-trained weights being provided. We denote these networks as MobileNet 0.25, 0.5, 0.75 and 1.0, respectively. α also affects the dimensionality of the extracted features from the network, which are the output of the final average pooling. The features are of length 1024, 768, 512 and 256 for values of $\alpha = 1.0, 0.75, 0.5$ and 0.25, respectively.

The training hyperparameters we use are those that perform best across several experiments. We train MobileNet 0.25 for 29 epochs and the rest of MobileNets for 39 epochs on Market-1501, by using SGD with a batch size of 32, an initial learning rate of 0.01 with a decay of 0.1 every 20000 steps and momentum of 0.9. On DukeMTMC-reID, we train all MobileNets for 39 epochs. We set the initial learning rate to 0.01 for MobileNet 0.25 and to 0.02 for MobileNet 1.0. For MobileNets 0.5 and 0.75 we use a batch size of 16 and a starting learning rate of 0.005.

Network distillation

ResNet-50 plays the role of the teacher as it is the largest network among those considered. However, we also consider MobileNet 1.0, which has the biggest capacity of the MobileNets configurations. The number of parameters for MobileNets are 4.24M, 2.59M, 1.34M and 0.47M for width multiplier values of 1.0, 0.75, 0.5 and 0.25 respectively, while ResNet-50 has 23.5M of parameters. Aiming at an efficient network, the student is the MobileNet with the smallest width multiplier, *i.e.* MobileNet 0.25.

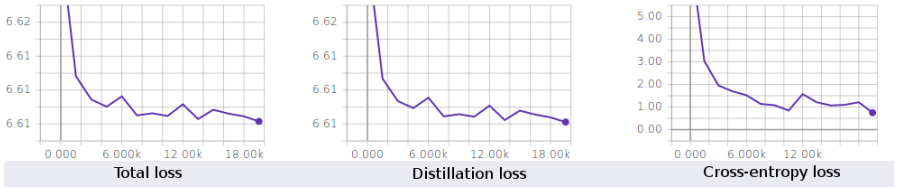
We analyze the effect of the hyperparameters of the distillation, *i.e.* the temperature T and the regularization weight λ for the distillation loss (Eq. 2.3). We explore the range of temperatures $T \in [1-30]$, being $T = 1$ the case in which the entropy of the soft targets is not modified and $T = 30$ a case of very high temperature. This selection is based on the observed softened probability distribution that is generated by the teacher network for $T = 30$, as shown in Figure 2.5. In that probability distribution the difference between the probabilities assigned to the incorrect classes and the one assigned to the correct class is less than a 0.1%. This is due to a very high temperature that causes the probability distribution to be almost flat and corresponds to the case of maximum entropy. To do the analysis for T in that range, we use intervals of 5, and 1 for the lowest values. For λ, we choose the values 0.0001, 0.001 and 0.01. They have been selected by analyzing the contribution of the loss terms while monitoring the training process, as shown in Figure 2.6. When using a value of $\lambda = 0.1$, the cross-entropy loss leads the training and the distillation term barely affects. In this situation however, our experiments showed this makes the training harder to converge, resulting in a performance drop.

For this reason, we do not consider values of $\lambda \geq 0.1$ in our analysis.

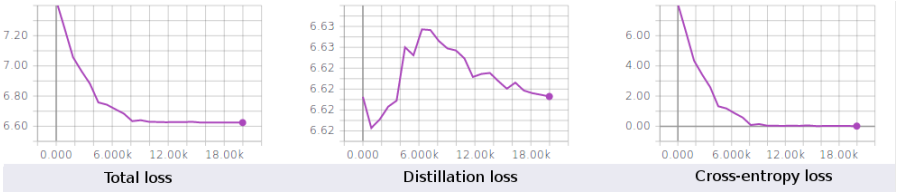
Figure 2.5: Original and softened probability distributions generated by the teacher network for temperature values of (from top to bottom) T=1,3,5,10,20,30.

For each value of T, we evaluate both the Rank-1 accuracy and mAP with the features extracted from the student network. We try several combinations of the hyperparameters, *e.g.*, learning rate, batch size, number of epochs. However, most

of the experiments perform best using the same hyperparameters, *i.e.*, we obtain that the same optimum configuration of parameters for several values of T and λ. All the Rank-1 and mAP values reported in Section 2.6 for each value of T, are those that perform best among all the experiments performed. Most of the distillation experiments use SGD, with an initial learning rate of 0.02 that decays 0.1 every 20000 steps, and a momentum of 0.9, being trained for 39 epochs.



(**a**)



(**b**)

Figure 2.6: Training loss for the distillation with (**a**) low ($\lambda = 0.0001$) and (**b**) high ($\lambda = 0.1$) λ values. The distillation loss leads the training in case (**a**) , while in case (**b**) it is done by the cross-entropy loss (see Eq. 2.3).

2.6 Results

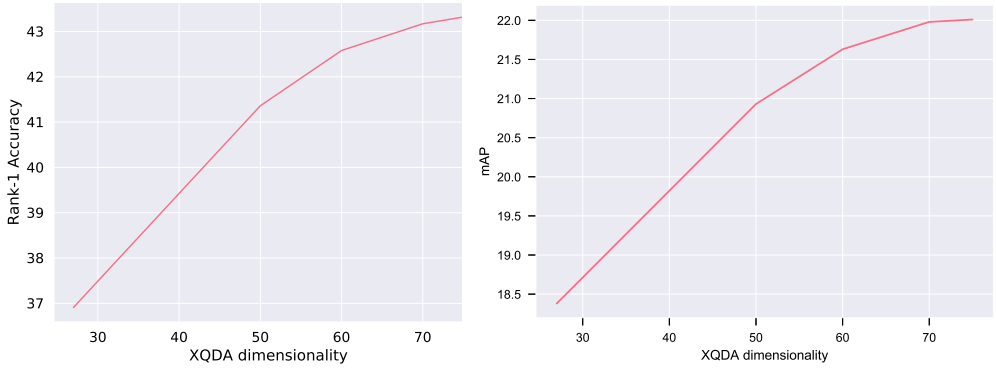
Hand-crafted features

For the classical approach using LOMO and XQDA, we report its performance in Table 2.1. The results verify using metric learning algorithms such as KISSME or XQDA significantly improves the overall performance over the standard Euclidean distance. However, we must take into account that in these experiments PCA is previously applied in the case of KISSME to reduce the dimensionality of the LOMO features to 200. The dimensionality in the XQDA space is 75, which is considerably smaller. XQDA then performs better than KISSME even with a stronger dimensionality reduction.

However, both XQDA and KISSME require a metric learning step that increases the computational cost. In particular, the XQDA training, *i.e.* finding the projection matrix from the training set samples, takes 892 seconds for Market-1501, whose training set contains 12936 images. Also, comparing a query image against the gallery takes an average time of 1,951 ms per image. Using XQDA, the system compares the individuals' features at a rate of 0.5 images/s. Regarding the computational cost for feature extraction with LOMO, the mean CPU time to extract the LOMO features per image is 17.5ms, *i.e.* the system is able to get the descriptors for the images of the individuals at a rate of 57 images/s.

Table 2.1: LOMO and XQDA performance on Market-1501.

Features	**Similarity metric**	**Rank-1** (%) ↑	**mAP** (%) ↑
LOMO	Euclidean distance	27.11	8.01
LOMO	KISSME [42]	41.83	19.37
LOMO	XQDA (dimensionality 75)	43.32	22.01



(**a**) (**b**)

Figure 2.7: Performance for LOMO + XQDA on Market-1501 depending on the XQDA dimensionality. (**a**) Rank-1 accuracy and (**b**) Mean average precision.

Figure 2.7 shows the dependency of the performance with the XQDA dimensionality, while on Table 2.1 the performance reported of LOMO+XQDA corresponds

to the highest dimensionality value for XQDA. The accuracy increases with the dimensionality of XQDA, as more information can be encoded in the feature vector. Although we expect a saturation on the performance from a certain value, we do not reach such value. This is probably because the maximum dimensionality in our case is 75, which is considerably low. It is much lower than the dimensionality of the smallest feature vectors considered in our experiments, that is 256 for MobileNet 0.25.

Deep features

For the deep features baseline, [122] get a 72.54% of rank-1 accuracy and 46% mAP on the Market-1501 dataset, with deep features extracted from ResNet-50. Following the same strategy, in [128] the baseline results for the DukeMTMC-reID dataset are a 65.22% of rank-1 accuracy and 44.99% of mAP.

Table 2.2: Rank-1 accuracy, mean Average Precision (mAP) and computational cost of the inference for the deep features from the ResNet-50 and MobileNet architectures trained on the Market-1501 and DukeMTMC-reID datasets.

Market-1501	**Rank-1** (%) ↑	**mAP** (%) ↑	**# images/s** ↑
ResNet-50	64.46	38.95	128
MobileNet 0.25	59.74	34.13	613
MobileNet 0.5	68.11	41.52	607
MobileNet 0.75	67.34	40.44	574
MobileNet 1.0	67.37	39.54	545
DukeMTMC-reID	**Rank-1** (%) ↑	**mAP** (%) ↑	**# images/s** ↑
ResNet-50	67.1	44.59	128
MobileNet 0.25	49.69	28.67	613
MobileNet 0.5	54.62	32.17	607
MobileNet 0.75	57.32	34.69	574
MobileNet 1.0	57.41	34.86	545

Table 2.2 shows the performance of ResNet-50 and MobileNets fine-tuned to the target datasets. On Market-1501, the middle size MobileNets perform best, even slightly better than the biggest one and ResNet-50. However, MobileNet 0.25 achieves a lower performance. The reason why the middle models perform so well could be that all of them have enough capacity to solve the problem. Then, a bigger architecture, such as ResNet-50, does not involve an improvement. Moreover, as mentioned in Section 2.5.3, training the networks on a dataset with a high number

of classes and a small number of samples per class is not straightforward. The baseline achieved with ResNet-50 by [122] suggests a higher performance could be achieved for this network.

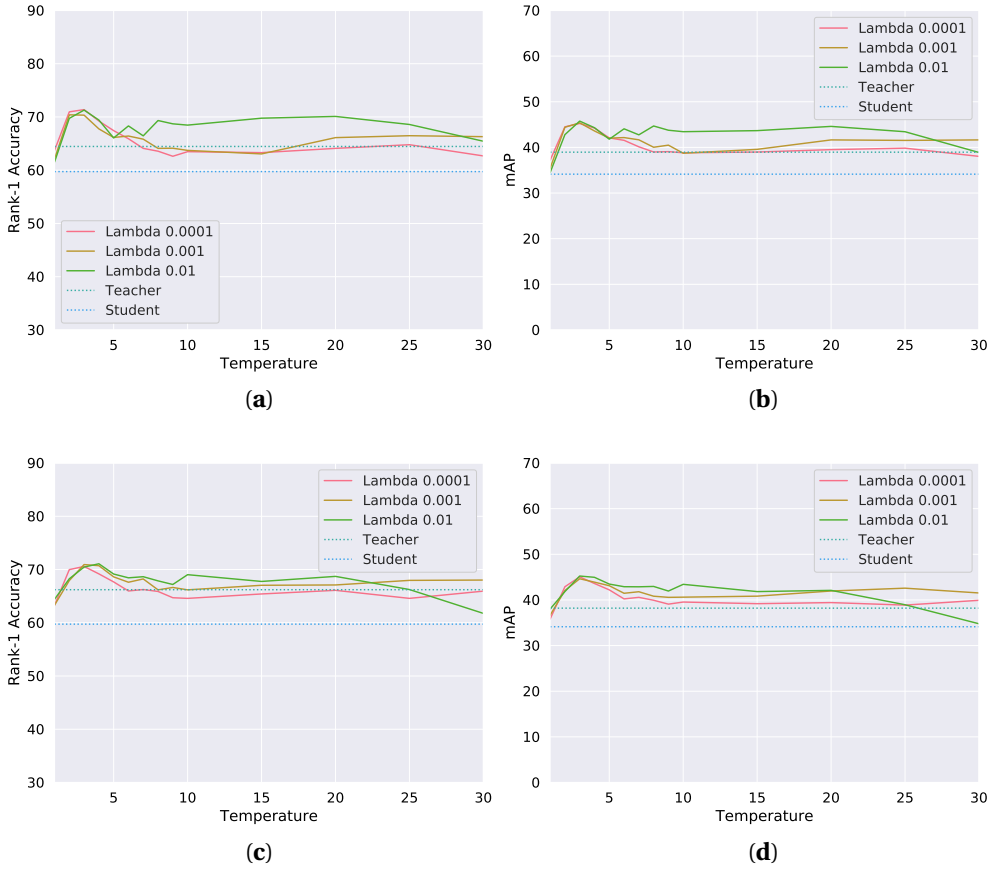


Figure 2.8: Distillation performance on Market-1501. (**a**) Rank-1 accuracy and (**b**) Mean average precision for student model MobileNet 0.25 with teacher model ResNet-50. (**c**) Rank-1 accuracy and (**d**) Mean average precision for student model MobileNet 0.25 with teacher model MobileNet 1.0. Best viewed in colour.

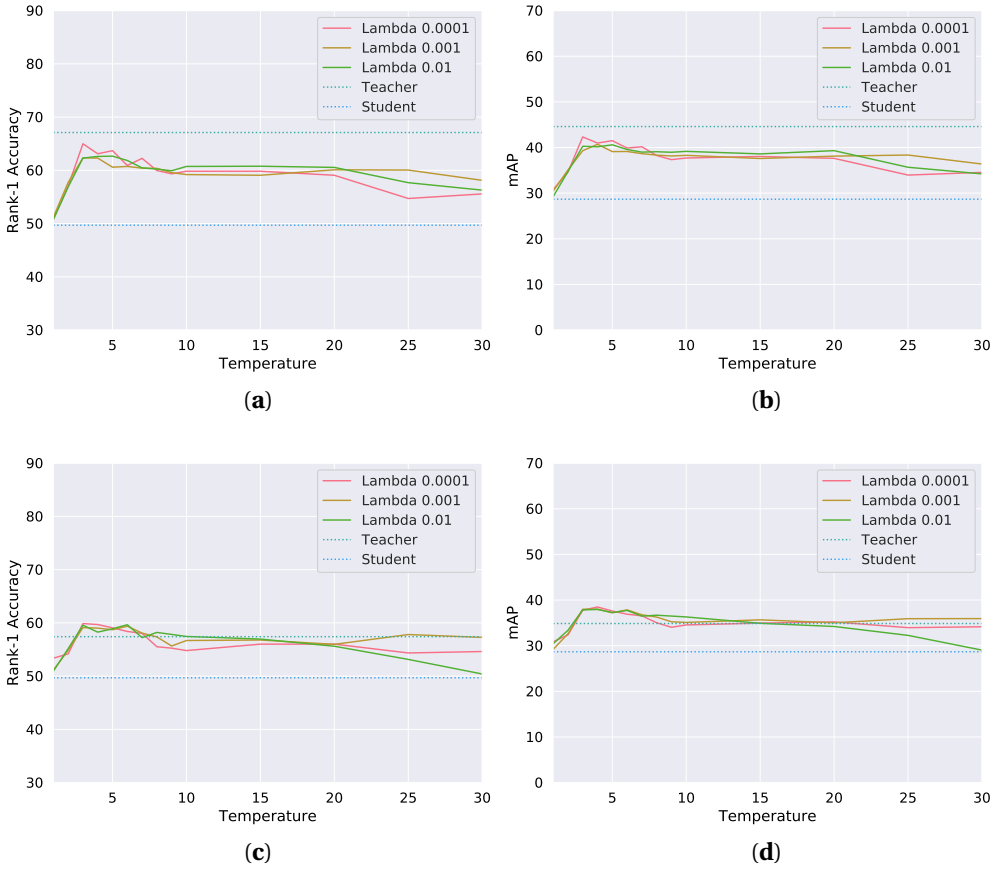


Figure 2.9: Distillation performance on DukeMTMC-reID. (**a**) Rank-1 accuracy and (**b**) Mean average precision for student model MobileNet 0.25 with teacher model ResNet-50. (**c**) Rank-1 accuracy and (**d**) Mean average precision for student model MobileNet 0.25 with teacher model MobileNet 1.0. Best viewed in colour.

For the DukeMTMC-reID dataset, MobileNets do not perform as good as they do for Market-1501. The reason might be this dataset is more challenging, and requires a higher capacity of the network to provide a good enough description of the identities. Since the size of the bounding boxes vary and all of them have to be resized to 128×128, losing thereby the aspect ratio, the input images have a higher

variability.

Network Distillation

The network distillation experiments are carried out by using pre-trained ResNet-50 and MobileNet 1.0 networks as teachers, whose performance is reported in Table 2.2. We show in Figures 2.8 and 2.9 the Rank-1 accuracy and mAP dependency with the temperature in the distillation, for the Market-1501 and DukeMTMC-reID datasets, respectively. The performance of the teacher and the student trained independently is also provided in the previous figures to show the comparison w.r.t. the baseline without distillation. All the experiments improve significantly the performance of the student, and even outperform the teacher at low temperatures. The only case in which the student does not outperform the teacher is for the DukeMTMC-reID dataset when the teacher network is ResNet-50 (Figure 2.9 (a,b)). In this case, however, the difference of performance between the teacher and the student is higher than for the other experiments.

For a fixed value of λ, there is always a peak of performance in $T = 3$. The worst performing temperature value is for T=1. This corresponds to the case in which the temperature is not increased, *i.e.* the original logits from the teacher models are used. This demonstrates the importance of raising the temperature to produce suitable soft targets. Also, from a certain value of T, the performance gets saturated, probably because the probabilities are already very softened and they do not change significantly for higher values of T, as Figure 2.5 shows (values of $T = 20, 30$). The differences of probabilities among both distributions are less than a 0.1%.

Table 2.3 compares our best performing configuration for network distillation to the state-of-the-art approaches. Our proposed approach is not superior than the state of the art in terms of performance. However, we must consider that it is intended to be efficient and simple. The specific design towards an efficient solution can compromise the accuracy. Also, while we only naively train a classification network with standard cross-entropy and network distillation, state-of-the-art approaches employ more complex architectures. For instance, [30] include generative adversarial network in their solution. The work in [85] leverages attributes information that needs to be previously detected automatically. A decorrelation step applied on the weight vectors of the last fully-connected layer is required in [94]. All of these approaches are presumably less efficient due to their increased complexity. The only work that is comparable in terms of efficiency is [116], that also employs network distillation on the same networks as ours. Their learning objective, however, is more sophisticated than ours, which might explain the gap in performance.

Finally, to summarize all the considered methods, Figure 2.10 and Table 2.4 show

Table 2.3: Rank-1 accuracy and mean Average Precision (mAP) for network distillation, taking MobileNet (α =0.25) as the student network, and MobileNet (α =1.0) and ResNet-50 as the teachers, compared against the state of the art on the Market-1501 and DukeMTMC-reID benchmarks. * These methods were not yet published when this work was developed. We added them according to reviewers suggestions.

Market-1501	**Rank-1** (%) ↑	**mAP** (%) ↑
MobileNet 0.25 distilled from ResNet-50	71.29	45.76
MobileNet 0.25 distilled from MobileNet 1.0	70.46	45.24
P2S [132]	70.72	44.27
CADL [58]	73.84	47.11
MSCAN Fusion [52]	80.31	57.53
SVDNet [94] *	82.3	62.1
ACRN [85]	83.61	62.60
DML [116] *	89.34	70.51
FD-GAN [30] *	90.5	77.7
DukeMTMC-reID	**Rank-1** (%) ↑	**mAP** (%) ↑
MobileNet 0.25 distilled from ResNet-50	64.99	42.32
MobileNet 0.25 distilled from MobileNet 1.0	59.69	38.48
Dataset baseline with ResNet-50 [128]	65.22	44.99
ACRN [85]	72.58	51.96
SVDNet [94] *	76.7	56.8
FD-GAN [30] *	80.0	64.5

the trade-off between computational cost and accuracy. In Table 2.4 we compare the performance of the classical approach (LOMO+XQDA), the deep features extracted from the MobileNets architectures trained with the cross-entropy loss as well as the deep features extracted from MobileNet 0.25 being distilled from the MobileNet 1.0 and ResNet-50 models. On the Market-1501 dataset, we compute the LOMO features then applying XQDA with dimensionality 75. The results for the DukeMTMC-reID dataset instead are reported in [128].

Note that LOMO computational cost is measured in CPU time, while all the deep features methods are measured in GPU time. The comparison of the computational cost is therefore not strictly fair. In terms of accuracy, the LOMO+XQDA accuracy is the lowest by a large margin, as expected for a hand-crafted method. This kind of method would be suitable only for an application in which either a GPU, or large amounts of annotated data, are not available. The results show distillation

Table 2.4: Evaluation of the trade-off between Rank-1 accuracy, mean Average Precision (mAP) and computational time on the Market-1501 and DukeMTMC-reID datasets. *d.f.* stands for *distilled from*.

Market-1501	**Rank-1** (%) ↑	**mAP** (%) ↑	**# images/s** ↑
LOMO + XQDA	43.32	22.01	57
ResNet-50	64.46	38.95	128
MobileNet 1.0 independent	67.37	39.54	545
MobileNet 0.25 independent	59.74	34.13	613
MobileNet 0.25 d.f. ResNet-50	71.29	45.76	613
MobileNet 0.25 d.f. MobileNet 1.0	70.46	45.24	613
DukeMTMC-reID	**Rank-1** (%) ↑	**mAP** (%) ↑	**# images/s** ↑
LOMO + XQDA [128]	30.75	17.04	57
ResNet-50	67.1	44.59	128
MobileNet 1.0 independent	57.41	34.86	545
MobileNet 0.25 independent	49.69	28.67	613
MobileNet 0.25 d.f. ResNet-50	64.99	42.32	613
MobileNet 0.25 d.f. MobileNet 1.0	59.69	38.48	613

effectively improves the performance of efficient networks, providing the best accuracy among all the considered methods, as well as the lowest inference time. It is also worth mentioning the gap of computational cost between ResNet-50 and MobileNets, while their performance in terms of accuracy is very similar. This highlights the importance of choosing a suitable architecture for the target problem. For Market-1501, a network of the size of MobileNet can describe the features of the identities effectively while in the case of DukeMTMC-reID, ResNet-50 performs much better.

2.7 Conclusions

In this chapter we have addressed the problem of person re-identification aiming at an efficient solution. In a real-life application the working images show crowded scenes frequently, which justifies the need of having a system that is able to identify as many individuals as possible in the shortest time. We have considered both classical (*i.e.* not deep learning based) as well as current state-of-the-art deep learning based approaches, as we argue both could be suitable depending on the data and resources availability.

We have evaluated the trade-off between accuracy and computational cost for both kind of methods. As a classical approach, LOMO and XQDA comprise the two stages of image description and similarity metric computation. The deep learning based method employs features that are extracted from ResNet-50 and MobileNets networks and compared by using the Euclidean distance as the similarity measure. This evaluation is performed on large-scale person re-identification datasets, aiming to simulate the scenario of a real-world application. We showed that using features from CNNs outperformed by a large margin the accuracy achieved with a classical approach, being also much faster, when using a GPU. However, this requirement as well as the large amount of annotated data that a network needs to be properly trained are the drawbacks to consider. Both ResNet-50 and MobileNets achieve a good performance, being the latter four times faster at test time.

Additionally, we have proposed and discussed network distillation as an alternative to improve the performance of MobileNets at test time, demonstrating its effectiveness. The student MobileNets networks even outperform the teacher ResNet-50 model, getting an accuracy that could not be achieved by training the student independently.

(**a**)

(**b**)

Figure 2.10: Trade-off between the mean average precision (mAP) and the feature extraction time for the proposed methods on the (**a**) Market-1501 and (**b**) DukeMTMC-reID datasets. Note that the feature extraction time for LOMO is measured as CPU time while the deep features experiments are run on a GPU. Best viewed in colour.

3 Hierarchical Novelty Detection

3.1 Introduction

Deep neural networks have demonstrated to achieve outstanding performance on image classification. However, the problem of detecting samples that do not belong to any class known by the model, *i.e.* novelty detection, remains unsolved. Two challenges of this task are that, first, classification networks trained by cross-entropy tend to be overconfident about their predictions, meaning they will assign a known class to any input fed to the network with very high confidence. The second difficulty is that by definition there is no training data for what is *novel*. There have been some efforts in addressing such problem [34, 48, 49, 66], but the binary output of these approaches only determines whether the sample belongs to a known class or is unknown. A desirable feature of classifiers would be, besides providing a novel/known decision, to produce an approximate prediction of the novel class by taking advantage from the knowledge of the already learned classes. In particular, we go beyond vanilla novelty detection and study how to perform such enhanced novelty detection under the framework of a hierarchical taxonomy of classes. This problem is known as hierarchical novelty detection [47]. It aims at correctly classifying samples of known classes, while also allocating the novel samples to the most suitable node of the hierarchy, *i.e.* their parent class. Figure 3.1 illustrates a simplified example. Let us assume a model trained on traffic sign recognition that has learned to only recognize speed limit traffic signs of 10, 20, 50, 90 and 120. If the system is fed a sample image of a 30 speed limit, it then should predict this sample belongs to a novel class, and more precisely, that it is a *speed limit* sign.

This problem has been traditionally studied as two independent tasks in the literature, *i.e.* novelty detection and hierarchical classification. Solving the joint task, however, has the advantage that novelty detection can benefit from the hierarchical taxonomy of classes. In academia, experiments are often restricted to certain sets of classes that compose the datasets. This data is limited and cannot comprise all the possible classes and variability of samples a real-life application faces. These ex-

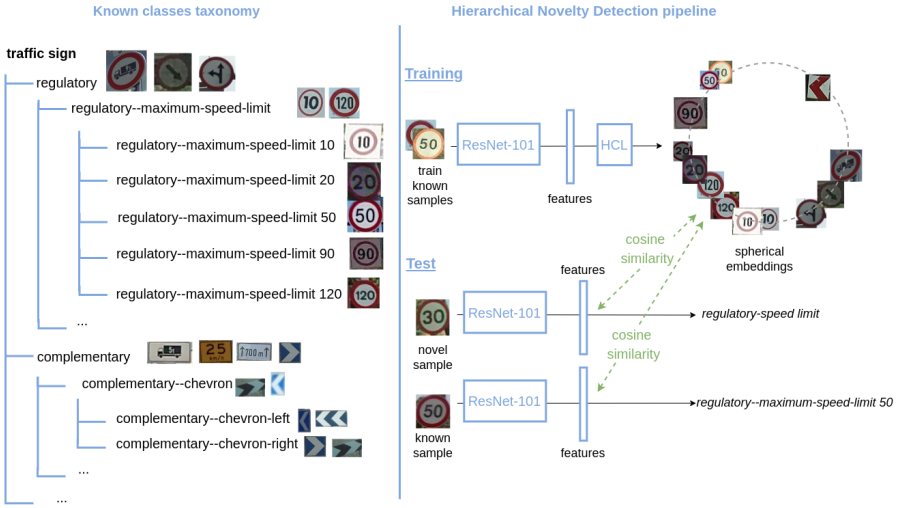


Figure 3.1: Example of hierarchical novelty detection on traffic sign recognition. Our system is trained to recognize speed limit traffic signs of 10, 20, 50, 90 and 120. When it is fed an image of a speed limit traffic sign of 30, it should predict that it belongs to a novel class (never seen during training), but also that is a *speed limit* traffic sign, placing correctly the novel sample in the hierarchical taxonomy of known classes.

periments, moreover, are based on a closed-world assumption [12, 26], *i.e.* systems consider the only existing classes are those seen at training time. Novelty detection instead, necessarily considers an open world setting. By leveraging the hierarchical taxonomy of the known classes, we show it is possible to produce approximate predictions even for unknown samples, by classifying them to the closest concept in such semantic taxonomy.

In this chapter, we propose to solve the problem of hierarchical novelty detection by introducing a novel loss function, *i.e.* Hierarchical Cosine Loss (HCL), which learns an embedding of discriminative features that is consistent with the taxonomy class relationships by encoding taxonomy based constraints. In this embedding, every known class, that corresponds to either a parent or leaf node, is represented by a *prototype*. *Prototypes* enable the classification of any kind of sample, including novel ones. HCL is based on a normalized version of the softmax loss reformulated from a cosine perspective. It optimizes the cosine similarity at training time between features and corresponding class *prototypes*. Consistently, we perform the novelty

decision at test time by using the same metric. By mapping the sample into the embedding space, our approach assigns the sample features to the *prototype* with the highest cosine similarity.

To the best of our knowledge, there is only a previous work [47] that has approached this problem. The authors instead employ confidence calibrated classifiers [48] to overcome the difficulty of the overconfidence of models trained with standard softmax. Similar approaches to ours [99] have been proved to increase the performance on the face recognition task in comparison to the standard softmax formulation that [47] applies. Whereas the mathematical background is similar, there is a subtle conceptual difference among both formulations. While standard cross-entropy answers the question *What is this sample?*, ours tries to find the response to *What does this sample look like most?* The latter setting seems to be more appropriate to classify unknown samples by finding the most similar known class.

Our solution can be a powerful tool for practical applications. For instance, a potential application is to ease annotation procedures, that could be semi-automated by providing the closest known class even for novel samples. Also, it could be a first step towards class incremental learning [49, 131], where one could extend the model with the newly learned classes. As a concrete application, autonomous driving technologies can benefit from it, *e.g.* by detecting new object categories automatically in a navigation system and suggesting the most similar known class. The aforementioned class incremental setting could also be used to build adaptive models to the challenging changing environment that autonomous driving systems face. We specifically address the traffic sign recognition problem [98]. This task is of special interest because in the case of traffic signs, the semantic taxonomy is strongly related to the visual appearance. The categories are human-built so that the meaning is intended to be visually represented. Therefore, one could build an adaptive traffic sign detector that is able to infer the meaning, at least partially, of the detected novel signs.

In summary, the contributions of this chapter are the following:

- A hierarchical novelty detection framework that is able to detect novel samples that belong to classes not seen during training, also placing them at the correct node of the taxonomy, *i.e.* predict the parent class. For this purpose, we introduce a novel loss function, *i.e.* Hierarchical Cosine Loss, which incorporates hierarchical constraints and optimizes the cosine similarity as the confidence metric, differently from most of current approaches that are based on class probabilities.

- A specific application to traffic sign recognition. We introduce the taxonomies and appropriate splits for two large scale traffic signs datasets, Mapillary

Traffic Sign Dataset (MTSD) and Tsinghua-Tencent 100K (TT100K).

- We show that HCL significantly outperforms state-of-the-art approaches on these traffic sign benchmarks. For TT100K and MTSD, our method can detect novel samples from unknown classes at the correct nodes of the hierarchy with 75% and 24% of accuracy when it correctly classifies known classes with 90% of accuracy, respectively. It also reaches 81% and 36% of novel accuracy at 80% known accuracy for TT100K and MTSD, respectively. Additionally, on the natural images datasets AWA2 and CUB, it achieves equivalent performance to state-of-the-art models.
- A new hierarchical novelty detection metric, *i.e.* the average error distance $\bar{d}_h$, to evaluate the errors produced under a hierarchical setting. It measures how far in the hierarchy we predict novel classes from the correct node.
- An ablation study that analyzes the individual performance of the HCL terms, discussing their benefits and drawbacks.

3.2 Related Work

3.2.1 Novelty Detection

Broadly, novelty detection belongs to the field of study of out-of-distribution detection [28, 49, 66, 71], that consists on identifying samples that do not belong to the distribution of the training data (in-distribution). More specifically, novelty detection aims to classify known classes while detecting *novel* samples that correspond to classes never seen during training. For instance, the authors of [66] address out-of-distribution detection by proposing a metric learning based approach. They distinguish among *novelties* and *anomalies* depending on the resemblance w.r.t. the in-distribution data. Similarly to us, they apply it to traffic sign recognition. However, they only provide a binary output that classifies a sample into either a known class or a generic class of *novelty*. Our approach instead, provides information about what kind of *novelty* it is, by predicting its parent class as the expected output. In a different direction, [49] considers both out of distribution detection and adversarial attacks, as both problems consist in detecting abnormal samples. They propose a Gaussian discriminant analysis resulting in a confidence score based on the Mahalanobis distance. Moreover, they apply their approach into a class incremental setting framework, showing they are able to incorporate new classes without retraining the models.

There are no works other than [47] directly addressing hierarchical novelty detection. The reason is probably it is a concrete and complex task that merges

two problems traditionally studied separately, *i.e.* hierarchical classification and novelty detection. The authors of [47] introduced the problem for the first time and proposed two different models. The first *Top-Down* model trains confidence-calibrated classifiers [48], which besides training the standard cross-entropy loss, minimize the KL divergence of the probability vector w.r.t. the uniform distribution. At test time, it makes top-down decisions so that at each node it measures the KL divergence to evaluate whether the classifier is confident about the prediction, which determines if the sample is novel or known when compared to a threshold. The second *Flatten* model trains the standard cross-entropy loss considering all classes, *i.e.* both leaf and super classes, and performs the decision ignoring the taxonomy. Additionally, they show the hierarchical embeddings can be employed to improve the performance on generalized zero shot learning. Both their proposed approaches employ the standard softmax objective and base their training and decision on class probabilities. Differently to them, we train our embeddings by optimizing the cosine similarity instead of the inner product and perform the novelty decision based on this similarity metric. Furthermore, our approach learns an embedding of discriminative features that is consistent with the taxonomy class relationships.

Nevertheless, there exist some problems that are conceptually similar. One of the closest problems is zero-shot learning (ZSL) [31, 87, 107, 108], where the goal is to classify samples of classes not seen during training. The base idea of hierarchical novelty detection, *i.e.* to use the knowledge of the known classes to recognize the novel ones, is shared with ZSL. It however, requires additional information about the known classes to be given, in the form of attributes or text description transformed into embeddings, while hierarchical novelty detection only relies on the class taxonomy.

3.2.2 Hierarchical Classification

Considering hierarchical class taxonomies in the classification problem has been widely studied in the literature [9, 14, 15, 17, 29]. The problem of hierarchical novelty detection actually comprises hierarchical classification of the known classes. In [9], the authors take advantage from hierarchical taxonomies of classes for error measurement. They propose two methods based on the cross-entropy loss that aim to minimize the asymmetric cost of the errors produced. Their error evaluation employs the height of the lowest common ancestor (LCA) among the predicted and the ground-truth classes in the taxonomy tree. This is similar to the metric we propose for the task of hierarchical novelty detection in Section 3.5.2, the average error distance $\bar{d}_h$. Differently, we use the distance in the tree between both classes, that corresponds to the sum of distances from both the predicted and ground truth

classes to their LCA. More recently, a *prototypical network* is introduced in [29], that is supervised by employing a cost matrix that encodes hierarchical relationships among classes, then penalizing large hierarchical errors. It is conceptually similar to our proposed loss in that they also incorporate hierarchical constraints to learn an optimal embedding. Also, they consider the *Average Hierarchical Cost* as a metric to evaluate classification errors, which matches the definition of our average error distance $\bar{d}_h$ metric, but in the context of hierarchical classification.

Another related problem is long tailed recognition, that consists in correctly classifying classes from which many of them are underrepresented in the training data. This often matches a real life scenario, where balanced data for all the classes is unlikely to have. The obvious differences are their classes are highly imbalanced but at least one sample per class is seen during training and they do not need to make a novel/known class decision. Some works employ class hierarchies in their solution. For instance, the authors of [106] propose to solve this problem under a hierarchical class taxonomy framework, then providing from coarse to fine-grained predictions according to the confidence. This enables the models to reject classification at different levels. More recently, [18] transform the problem into a hierarchical classification one by building a tree which levels correspond to different degrees of difficulty according to how imbalanced the data is, then transferring the knowledge across levels.

3.2.3 Cosine Losses

There exist diverse works that similarly to us, propose loss functions based on modifying the softmax loss from a cosine perspective to improve its performance. The softmax loss, in this context, refers to a cross entropy loss preceded by a softmax activation and a fully connected layer. These works are commonly applied to the face recognition task, where learning discriminative features is essential to distinguish identities. They also benefit from this loss formulation because they use the cosine similarity at test time.

The first work that opened this line of research was [62], where based on the softmax loss, the proposed L-Softmax loss included a new angular margin hyperparameter that acts on the class decision boundaries to enforce inter-class variance then pushing the discriminative power of the features. SphereFace [61] normalized the weights of the last fully-connected layer on the L-Softmax loss, making them lie on a hyphersphere. A normalized version of the softmax loss was introduced in [99], where they normalized both features and class weights so that the only variable to be optimized is the cosine of the angle between them. Later, the authors of [100] added a margin parameter to it to increase the discriminative power of features. This margin separates the decision boundary between classes in the embedding

space, at the cost of introducing a new hyperparameter. In our approach instead, to learn discriminative features under a hierarchical setting, we propose additional terms to the loss that encode hierarchy based constraints, being consistent with the problem we aim to solve. Similarly to [100], [24] also introduced a margin hyperparameter, but applied on the angle. Finally, in an effort to improve the aforementioned methods, AdaCos [115] proposes a hyperparameter-free approach, leveraging a dynamically adaptive scale parameter that is adjusted automatically. Simultaneously, RegularFace [119] proposed an *exclusive regularization* term to the loss to further push inter-class discriminability by optimizing angular distance among classes.

3.3 Hierarchical Novelty Detection

In this Section, we first describe the setting of the hierarchical novelty detection problem in Section 3.3.1, then introducing our proposed Hierarchical Cosine Loss in Section 3.3.2.

3.3.1 Class taxonomy

In hierarchical novelty detection, the classes are organized by a hierarchy of known classes that is built based on their semantics. The resulting taxonomies of the datasets considered in this chapter are trees, where all nodes have at least two children classes and a single parent. As an example, Figure 3.1 shows a subset of the taxonomy of MTSD. The dataset is split into two sets of disjoint classes: *known* and *novel*. *Known* classes are used during training to learn an embedding, while *novel* classes are not included in the hierarchy; they are never seen during training and our goal is to predict the correct parent (known) class for the novel samples at test time.

Datasets provide samples for known leaf classes, however, our approach also needs sets of samples that represent the parent classes. To this end, we employ a relabeling strategy as in [47]. We select a percentage of the samples of the leaf classes to be relabeled as their parent class. We refer to this percentage as the relabeling rate r_{rate}. This procedure is recursively repeated in a bottom-up manner from the bottom nodes to their parents until we reach the root and all the nodes are assigned samples. The subset of samples is chosen randomly and is different for each epoch.

3.3.2 Hierarchical Cosine Loss

We introduce the Hierarchical Cosine Loss (HCL) in order to learn an embedding for the known classes. HCL comprises a layer of learnable parameters that corresponds to a fully connected layer with no bias. The HCL layer is appended after the feature layer of a ResNet-101 backbone, which serves as a feature extractor. Our loss, HCL, is composed by a set of terms that enforce learning discriminative features, leveraging the class hierarchy. It is defined as follows,

$$HCL = \lambda_{NS} L_{NS} + \lambda_{HC} L_{HC} + \lambda_{CT} L_{CT} + \lambda_{HT} L_{HT}, \tag{3.1}$$

where L_{NS}, L_{HC}, L_{CT} and L_{HT} stand for Normalized Softmax, Hierarchical Centers, C-triplet and Hierarchical Triplet loss, respectively, and λ_{NS}, λ_{HC}, λ_{CT} and λ_{HT} are their regularization parameters.

Normalized Softmax Loss L_{NS}. A reformulation of the softmax loss was introduced in [99], consisting in applying normalization on both the weights from the last fully-connected layer, whose bias is set to 0, and the feature vectors. This results in optimizing the cosine similarity instead of the inner product. We refer to this loss as Normalized Softmax Loss (NSL). NSL is defined as

$$L_{NS} = \frac{1}{N} \sum_i -log \frac{e^{s\cos(\theta_{y_i,i})}}{\sum_j e^{s\cos(\theta_{j,i})}}, \tag{3.2}$$

where y_i is the ground truth label of the i-th sample, N is the number of samples and $\theta_{j,i}$ is the angle between W_j and x_i, being W_j a weight vector of the fully-connected layer for the j-th class and x_i the feature vector of the i-th sample. A weight vector W_j can be interpreted as a representative vector of the j-th class, we refer to it as a class *prototype*. For a class which features are properly separated in the embedding space, its *prototype* would correspond to the mean of the features. By applying L_2 normalization, we fix $\|W_j\| = 1$ and $\|x\| = s$. This results in optimizing only the cosine of the angle, as the norms will not contribute to the loss. After normalization, the feature vectors lie on a hypersphere, where the scaling parameter s controls its radius and the resulting features are separable in the angular space, reducing intra-class angular variability and pushing inter-class variance within the hypersphere. This consequently enforces removing radial variations.

C-Triplet loss L_{CT}. This re-formulation of the softmax loss can also be translated into the Constrastive or Triplet losses, which inspired us to propose the following loss terms that incorporate hierarchical constraints. In [99], the authors introduce the *C-triplet* loss $\mathscr{L}_{\mathscr{T}'}$ as the modified version of the triplet loss, that is defined as

follows,

$$\mathscr{L}_{\mathscr{T}'} = \max(0, m + \|\tilde{x}_i - W_j\|_2^2 - \|\tilde{x}_i - W_k\|_2^2), \quad \forall y_i = j, y_i \neq k, \tag{3.3}$$

where $\tilde{x} = \frac{x}{s}$ and m is a margin parameter. Note that both $\tilde{x}$ and W_j are normalized, then we could re-formulate it in terms of the cosine similarity. Considering that $\|\tilde{x}_i - W_j\|_2^2 = 2 - 2W_j^T \tilde{x}_i$ and $W_j^T \tilde{x}_i = \cos\theta_{j,i}$, $\mathscr{L}_{\mathscr{T}'}$ can also be defined as

$$\mathscr{L}_{\mathscr{T}'} = \max(0, m + 2\cos\theta_{k,i} - 2\cos\theta_{j,i}), \quad \forall y_i = j, y_i \neq k. \tag{3.4}$$

Then, considering pairs of different classes i, j, we define our C-triplet loss term L_{CT} as

$$L_{CT_i} = \max(0, \cos\theta_{j,i} - \cos\theta_{i,i} + m_{CT}), \quad \forall y_i \neq j. \tag{3.5}$$

where the margin parameter m_{CT} is set to zero in all our experiments for simplicity. This term is intended to increase the discriminative power of the features, increasing the inter-class variance. It encodes that the features of a class should be closer to their class than to other class centers, *i.e.* the cosine similarity is higher among the features of a class x_i and its *prototype* W_{y_i} than to the *prototypes* of different classes $W_j \mid y_i \neq j$.

Hierarchical Triplet loss L_{HT}. To further enforce discriminative features based on the hierarchical relationships, we propose the Hierarchical Triplet term L_{HT} that is defined as

$$L_{HT_i} = \max(0, \cos\theta_{k,i} - \cos\theta_{j,i} + m_{HT}), \quad \forall i, j, k \mid y_i \neq j \neq k,\ d_h(y_i, j) < d_h(y_i, k), \tag{3.6}$$

where d_h is the hierarchical distance between two nodes in the taxonomy, we refer the reader to Section 3.5.2 for more details on this distance. m_{HT} is a margin parameter and is set to zero in all our experiments. The purpose of this term is that features of a class should be closer to the *prototypes* of those classes that are closer in the taxonomy. For instance, a *speed limit* traffic sign class will be closer to any other *speed limit* sign than to any *direction* traffic sign. Figure 3.2 illustrates an example. The effect of this term is then to distribute the features in the hypersphere according to the taxonomical relationships.

Hierarchical Centers loss L_{HC}. Similarly to the Hierarchical Triplets term L_{HT}, the Hierarchical Centers loss L_{HC} aims to increase the separation in the angular space of the class *prototypes* W_j based on the hierarchical relationships between classes. The difference is that instead of being applied to the distance among

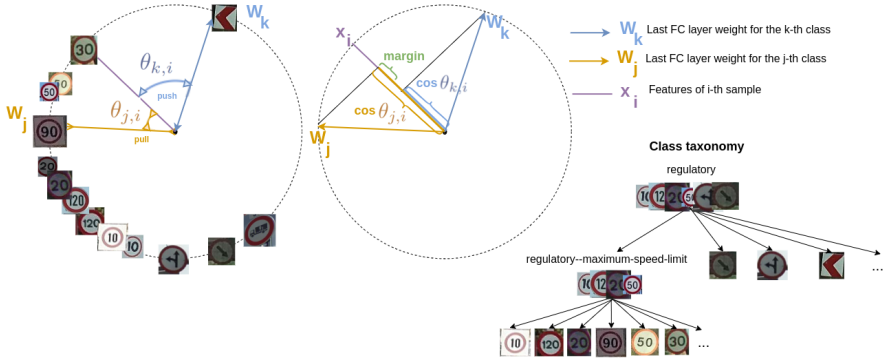


Figure 3.2: Interpretation of Hierarchical Triplet loss term L_{HT}. Samples are forced to be closer to *prototypes* of classes that are closer in the taxonomy. In this example, the anchor sample is a speed limit sign of 30, while the positive class is the speed limit sign of 90, which is closer in the taxonomy than the direction sign class, that is the negative class of the triplet.

features and *prototypes*, it only affects the class *prototypes* W_j. Thus, this term enforces a higher similarity among class *prototypes* that are closer in the taxonomy. L_{HC} is defined as

$$L_{C_i} = \max(0, \cos\phi_{y_i,k} - \cos\phi_{y_i,j} + m_C), \quad \forall i,j,k \mid y_i \neq j \neq k,\ d_h(y_i,j) < d_h(y_i,k), \tag{3.7}$$

where $\phi_{y_i,j}$ is the angle between W_{y_i} and W_j. m_C is a margin parameter and is set to 0.05 in all our experiments.

3.3.3 Inference

By training HCL for both known leaf and super classes, we learn the set of class *prototypes*, *i.e.* class weights W_j from the last fully connected layer that identify all the known classes. These, at test time, can be compared against the features of the test samples to perform classification.

At inference time, for every test sample we compute the cosine similarity between its features and all the class *prototypes* W_j. These features are extracted from the ResNet-101 model as we do at training time. We add an offset to the cosine similarities of the super classes, which controls the trade-off between known and novel class accuracies. Its value can be varied within a range to select the desired

working point. This is needed to compute the metrics detailed in Section 3.5.2. The test samples are finally classified to the class which *prototype* W_j has the highest similarity w.r.t. their features, after applying the offset. Then, if a sample is assigned a leaf class, it means it corresponds to a sample of this known leaf class, while if the sample is assigned a super class, it is considered as a novelty under this parent class. For instance, a sample classified as a *regulatory* traffic sign is a sign of an unknown class of type *regulatory* that in the taxonomy would be a child class of *regulatory*.

Note that, differently from other hierarchical classification methods [47], we do not follow a top-down strategy. This avoids top-down error aggregation that happens when the prediction at the top-most levels is wrong, and is magnified with complex and deep taxonomies. Therefore, we do the classification at inference time not considering any class taxonomy, being all the classes equally probable.

3.4 Datasets

We consider two kinds of datasets to assess the performance of our approach. First, to compare it against the state-of-the-art methods of hierarchical novelty detection, we employ the evaluation setting proposed in [47] as well as the datasets, CUB [103] and AWA2 [107]. Additionally, because we aim to apply our method on a traffic sign recognition framework, we choose two large scale traffic sign benchmarks: Tsinghua-Tencent 100K (TT100K) [133] and Mapillary Traffic Sign Dataset (MTSD) [27].

The original classes of these datasets are split into known and novel. Those that are known correspond to the leaf classes of the hierarchy. Among the samples of the known leaf classes, we build train, validation and test splits. Train samples are used to train the model, validation for hyperparameter optimization and test samples are used to evaluate the classification accuracy on the known classes. The details on how we make the splits are detailed in the following sections for each dataset. The data to reproduce our experiments is available in [79].

A dataset is more challenging as it has a larger number of samples, categories and has a more complex taxonomy of classes [23]. Table 3.1 contains this data for the datasets evaluated in this chapter. Note this information corresponds to the samples used in our experiments, where we have discarded some of the samples, and may differ from original benchmark statistics. Finally, the class taxonomies for these benchmarks are shown graphically in Appendix A.1.2.

Table 3.1: Datasets overview: number of samples, parent and leaf classes in the taxonomy tree and its height, for both known and novel splits. The height of a tree is the height of its root class, so that a tree of two levels is of height 1.

	Known				Novel	
Dataset	# Samples	# Parents	# Leaves	Height	# Classes	# Samples
AWA2	29408	21	40	5	10	7913
CUB	8814	43	150	5	50	2966
TT100K	21956	14	80	2	23	1735
MTSD	65312	40	164	3	39	4743

3.4.1 Tsinghua-Tencent 100K (TT100K)

The Tsinghua-Tencent 100K [133] dataset is one of the first large scale traffic sign benchmarks. It contains samples under different illuminance and weather conditions, extracted from real-life street view panoramas. In our experiments, we have used the cropped images of traffic signs. A difficulty of this dataset is that it is highly imbalanced. It is built from real-life images, where different traffic signs do not appear with the same frequency. Only 45 classes out of 221 have more than 100 examples, while the largest class has 2819 samples.

The criteria to decide which classes belong to the novel split is based on the number of samples per class. We first discard the classes with less than 10 samples to avoid errors. From the remaining classes, we take 20% of those least populated as novel, regardless of their position in the taxonomy. We think this split is the one that best simulates the data in a real world application, *i.e.* for a novel class the goal is to correctly classify its samples in the taxonomy, not needing many of them, while known classes should be properly learnt from a larger number of samples. The most logical option is therefore to select the classes with fewer samples as novel. To build the train/test splits, for each known class we keep 20% of the samples for test and within the remaining samples, 20% are used for validation.

Since this dataset has not been previously used for hierarchical novelty detection, we have built a taxonomy based on class semantics, *e.g.* for traffic signs of *prohibition limit of 20*, they should have a parent class that comprises *prohibition limit* signs at other speeds, while this class should have a parent class that comprises any kind of *prohibition* sign as well. A visual representation of the built taxonomy is shown in Appendix A.1.2.

3.4.2 Mapillary Traffic Sign Dataset (MTSD)

Recently, the Mapillary Traffic Sign Dataset has been introduced in [27]. It is the largest and most diverse traffic sign benchmark up to date. While TT100K contains only standard circular and triangular shaped signs, MTSD includes also direction, information or highway signs. Moreover, the images have been captured by multiple different camera devices all over the world. The benchmark provides fully and partially annotated traffic signs, although our experiments are restricted to only the fully annotated samples. Similarly to TT100K, MTSD is imbalanced despite having a larger number of samples per class.

The original class taxonomy of MTSD distinguishes as independent classes those that contain templates with the same semantics and similar appearance. However, we consider semantic based taxonomies, *i.e.* our application intends to classify the samples according to their meaning and not their appearance. For consistency we choose to merge different groups of templates that share the same semantics but are different in terms of appearance, into a single class, as shown in Figure 3.3. This increases the intra-class variability, but in exchange simplifies the taxonomy we would have if we distinguished these classes. From 313 classes in the original taxonomy, after merging those classes with same semantics, the resulting taxonomy has 203 leaf classes. Among these, 74 classes have less than 100 samples, while the largest class has 2775 samples.

Figure 3.3: Two examples of samples of MTSD that are distinguished as disjoint classes in the original benchmark because they share the same semantics but have a different appearance. Aiming at classifying according to semantics, we merge them as single classes. In the left case, we merge five classes into one (*complementary–chevron-left*) and in the right case, we merge four classes into one (*regulatory–no-parking*).

To build the hierarchical taxonomy, we create super classes that encompass the traffic sign categories provided in MTSD that share similar semantics, *e.g.* the classes *regulatory–no-left-turn* and *regulatory–no-right-turn* have a parent class *regulatory–no-turn*, that at the same time will have a *regulatory* parent class that comprises all the *regulatory* signs. Note that due to the different composition of classes of TT100K and MTSD, they do not share a unified traffic sign taxonomy.

There is no universal traffic sign taxonomy, to the best of our knowledge. It would be an interesting objective to explore in future work or a practical application, however.

Finally, we make the novel/known and train/test/validation splits by employing the same criteria and percentages as for TT100K.

3.4.3 AWA2, CUB

We employ the taxonomies of AWA2 and CUB provided in [47], which are built from the WordNet hierarchy by using its hypernym-hyponym relationships. Visual representations of the resulting hierarchies can be found in Appendix A.1.2. An interesting aspect of these taxonomies, differently to traffic signs, is that they obey to semantic hierarchical categories such as: *placental mammal* → *carnivore* → *canine* → *dog* → *shepherd dog*, where *carnivore* contains children classes as diverse as *bear* or *feline*. These high-level categories have no clear common features based on visual appearance. It is probably harder to learn how a *carnivore* looks like than how a *prohibition* sign looks like, since the concept is not reflected in the appearance but in deeper knowledge about what being a *carnivore* involves. These broad concepts are translated into a larger variability of samples under such conceptually high-level classes as well.

3.5 Evaluation

3.5.1 Experimental Setup

We compare our method against the state-of-the-art models proposed in [47]. They propose three models, from which we consider *TD+LOO* and *Relabel* for the sake of a fair comparison. *TD+LOO* is their best performing model while *Relabel* uses the same relabel strategy as ours to assign samples to parent classes. We run the implementation of these models provided by the authors.

To train our model, we consider two settings depending on the experiment. The first one consists in using fixed precomputed features by freezing the weights of the ResNet-101 backbone, while training the HCL fully connected layer to learn the *prototypes* W_j of the classes, as detailed in Section 3.3.2. In this setting we train HCL, but not the ResNet-101 backbone. In the second setting instead, we train jointly all the layers of ResNet-101 and HCL.

While carrying out the experiments, we noticed there was a moderate variability in the results even when using the same set of hyperparameters. Therefore, we repeat several times each experiment for a fixed set of hyperparameters. Instead of reporting the best-performing experiment from a set, the variability of the method

is worth to be analyzed. As we shall optimize our model to the validation set, a method whose performance is highly variable is not reliable, because we do not know how it will perform on the test set.

All our experiments are run on a set of GeForce GTX 1080, using multiple devices (at most four) in parallel when necessary, depending on the batch size.

3.5.2 Metrics

In order to assess the performance of our method on hierarchical novelty detection, we consider the following metrics. For comparison against the state-of-the-art approach proposed in [47], we employ the AUC of the novel/known accuracy curve and the novel accuracy at a fixed known accuracy point. In their work they select the point of 50% known accuracy as a reference. We use the average top-1 accuracy, so that a correct prediction is defined as follows, depending on the split. For known classes, their correct prediction is the ground truth label, while for novel classes, a correct prediction involves classifying it as the closest class in the taxonomy, *i.e.* its parent known class. The accuracy is averaged by the number of samples, independently of their label.

The novel/known accuracy curve is obtained by adding an offset to the similarity metrics of the potential novel classes, *i.e.* parent nodes. This offset value is varied so that we increase/decrease novel accuracy in detriment/favor of known accuracy, as both splits hold a trade-off relationship. The novel/known accuracy curve is built from a range of offset values that allows to explore all the available accuracy ranges. Accordingly, the AUC value is independent from the offset, *i.e.* it is independent from the working point.

On traffic sign benchmarks, we additionally consider points of interest at higher known accuracy points. In this context, we are interested in a working point in which our system classifies correctly most of the known classes, while performs as best as possible on the unknown ones. For this reason, the metrics that are more relevant are those of higher known accuracy points. In particular, we report the novel accuracy at 70% and 80% known accuracies, although we are interested in the range of known accuracy over 70%. For this reason, the AUC value is not a highly representative metric in our analysis, as it corresponds to the area for the full range.

Hierarchical error distance

The accuracy only evaluates if the prediction matches the correct label, but does not provide a measurement of the errors made. Specially under a hierarchical setting, we find this metric to be insufficient. Two wrong predictions of different degree of importance are treated as equally wrong by the accuracy. For instance, if the true

class of a sample is a *20 maximum speed limit regulatory* sign, predicting its class as a *10 speed limit regulatory* sign should be considered a smaller error than predicting it as a *chevron left complementary* sign. Figure 3.4 shows an alternative example. In fact, other works on hierarchical image classification [9, 29], stress the importance of optimizing error based metrics besides accuracy.

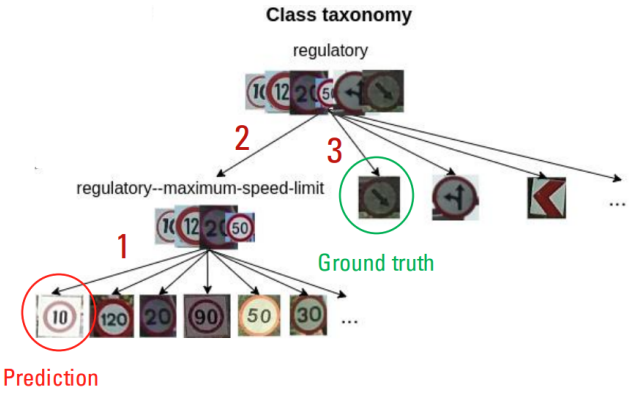


Figure 3.4: Example of the hierarchical error distance metric. For this sample, the value of the metric is equal to three, that corresponds to the shortest path in the tree among prediction and ground truth.

As a complementary metric to the accuracy, we introduce the hierarchical average error distance $\bar{d}_h$. It corresponds to the distance between the predicted and the correct class in the taxonomy tree. For the i-th sample, the hierarchical error distance $d_h(p_i, y_i)$ between the predicted class p_i and its ground truth label y_i is defined as the length of the shortest path in the tree, that corresponds to the sum of distances from both classes p_i and y_i to their lowest common ancestor (LCA). The average error distance $\bar{d}_h$ is then defined as

$$\bar{d}_h = \frac{1}{N} \sum_i d_h(p_i, y_i), \tag{3.8}$$

where N is the total number of samples. Note this distance metric is not normalized by the height of the taxonomy tree which affects its maximum value, *e.g.* in a taxonomy of 5 levels, the maximum error distance is 10, while in a taxonomy of 2 levels, it is 4.

In our experiments, we report the hierarchical average error distance for the novel split only, to analyze its dependency w.r.t. the accuracy of the known split.

This provides a measurement of the novelty detection error under such hierarchical setting.

3.6 Results and Discussion

Our experiments are divided into three parts. To evaluate the performance of our approach, HCL, in Section 3.6.1 we compare it to the state-of-the-art models in hierarchical novelty detection, *i.e. TD+LOO* [47] and *Relabel* [47]. In Section 3.6.1 we first consider the benchmarks where these models were originally evaluated, *i.e.* AWA2 and CUB. Then, in Section 3.6.1 we perform the evaluation on the target traffic signs benchmarks TT100K and MTSD. In the next sections we provide a more exhaustive evaluation of HCL on TT100K and MTSD. We compare the performance of different training strategies in Section 3.6.2. Finally, in Section 3.6.3 we analyze the individual contribution of each of the terms of HCL.

3.6.1 Comparison to State of the Art

AWA2 and CUB

For these experiments, we train HCL, *TD+LOO* and *Relabel* on top of features extracted from a ResNet-101 model that is only trained on ImageNet. This is the setting the authors of [47] chose, in their case claiming speed reasons. We use the exact hyperparameters and setting indicated by the authors. For HCL, the hyperparameters are chosen by optimizing them to the validation set (see Appendix A.1.1 for details on hyperparameters).

Table 3.2: Comparison of HCL against *TD+LOO* [47] and *Relabel* [47] on AWA2 and CUB. Performance is measured by the novel/known accuracy AUC and the novel accuracy and average hierarchical error distance $\bar{d}_h$ at 50% known accuracy. The reported values are the average from a set of 50 experiments $\pm 2\sigma$.

Method	AWA2			CUB		
	AUC	Novel acc @50% ↑	Novel $\bar{d}_h$ @50% ↓	AUC	Novel acc @50% ↑	Novel $\bar{d}_h$ @50% ↓
TD+LOO	25.7±4.3	33.1±6.1	1.82±0.23	18.0±1.0	9.9±1.1	2.56±0.10
Relabel	**33.7±5.9**	**38.7±6.8**	**1.65±0.12**	**28.9±2.2**	**38.1±3.5**	1.56±0.07
HCL	32.8±1.8	36.4±2.2	1.95±0.05	27.6±0.6	35.7±1.2	**1.34±0.03**

We report in Table 3.2 the metrics introduced in section 3.5.2 comparing the performance of our approach to *TD+LOO* and *Relabel*. The values of the metrics correspond to the average of 50 experiments and we provide an error of $\pm 2\sigma$. Fig-

ure 3.5 shows the novel/known accuracy trade-off and the average hierarchical error distance on the novel split over the known split accuracy. The dark curves of the plots correspond to the average of the set of 50 repeated experiments, while the shaded area around illustrates $\pm 2\sigma$ for each point.

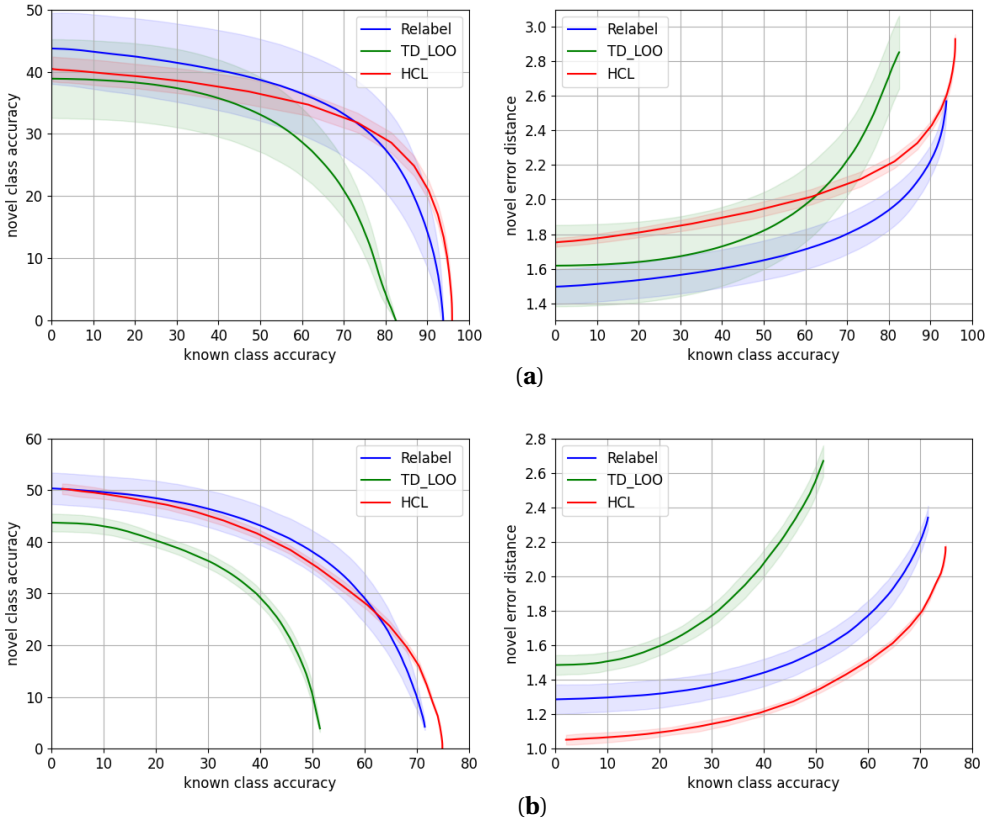


Figure 3.5: Novel/known accuracy trade-off and novel average hierarchical error distance over known accuracy, for HCL (red) and the state-of-the-art models *TD+LOO* [47] (green) and *Relabel* [47] (blue) for (**a**) AWA2 and (**b**) CUB.

AWA2. Considering only the mean of the experiments, *Relabel* [47] is superior in terms of accuracy in the range up to 70% known accuracy, while HCL performs better in the highest known accuracy range. This is the reason why *Relabel* gets the

highest AUC. However, if we take into account the variability of the methods, HCL and Relabel perform very similarly, *i.e.* their curves overlap except for the highest known accuracy range. Regarding the novel hierarchical error distance $\bar{d}_h$, *Relabel* [47] consistently makes smaller errors on the novel split.

CUB. In terms of accuracy, both *Relabel* and HCL perform similarly although the variability of *Relabel* is higher. *Relabel* performs better than the other variants up to ~ 60% known accuracy while HCL is superior in the uppermost known accuracy range. HCL makes a consistent smaller average error throughout all the accuracy ranges.

Our results show HCL performs similarly to the state-of-the-art methods [47] on the natural images benchmarks, AWA2 and CUB. It shows a slightly higher novel accuracy at the highest known accuracy ranges, while the errors made by the model are smaller than *TD+LOO* and *Relabel* on CUB but higher on AWA2.

TT100K and MTSD

Instead of using the features from a model trained on ImageNet as in the previous experiments, for TT100K and MTSD we find necessary to perform a fine-tuning of ResNet-101 using the cross-entropy loss. This is because traffic signs are a very specific kind of data, with a visual appearance different to ImageNet images. The comparison of performance when using features finetuned or not to the target dataset, will be discussed later in section 3.6.2.

Table 3.3: Comparison of HCL against the state-of-the-art models on TT100K and MTSD. Performance is measured by the novel/known accuracy AUC and the novel accuracy and average hierarchical error distance $\bar{d}_h$ at 50% and 70% known accuracies. The values are the average from a set of 50 experiments $\pm 2\sigma$.

Dataset	Method	AUC	Novel acc ↑		Novel $\bar{d}_h$ ↓	
			@50%	@70%	@50%	@70%
TT100K	TD+LOO [47]	42.2±2.6	55.8±4.6	12.6±1.5	0.68±0.09	1.47±0.20
	Relabel [47]	48.4±3.7	52.3±4.0	48.7±4.2	0.63±0.06	0.66±0.07
	HCL	**84.1±0.7**	**87.2±0.8**	**83.7±1.1**	**0.15±0.01**	**0.18±0.01**
MTSD	TD+LOO [47]	30.6±1.5	36.4±2.6	9.5±1.9	1.31±0.10	2.12±0.08
	Relabel [47]	27.3±3.8	30.9±5.0	24.6±3.6	1.28±0.11	1.44±0.09
	HCL	**44.2±1.6**	**47.7±2.1**	**40.5±2.3**	**0.78±0.04**	**0.89±0.05**

The fine-tuning is performed by training ResNet-101 for 1000 epochs using a batch size of 140 and a learning rate of $1 \cdot 10^{-4}$ with an Adam optimizer, for both datasets. Once ResNet-101 is trained, we extract the features to train *TD+LOO*, *Relabel* and our model, as we did for AWA and CUB. It is also possible to train

simultaneously the ResNet-101 backbone and HCL. However, we chose to do a separate fine-tuning to keep the setting proposed in [47] for the sake of a fair comparison. The fine-tuning was performed only once, while the experiments for HCL, *TD+LOO* and *Relabel* were repeated for 50 times with the set of best performing hyperparameters on the validation set. We refer the reader to Appendix A.1.1 for details on hyperparameters.

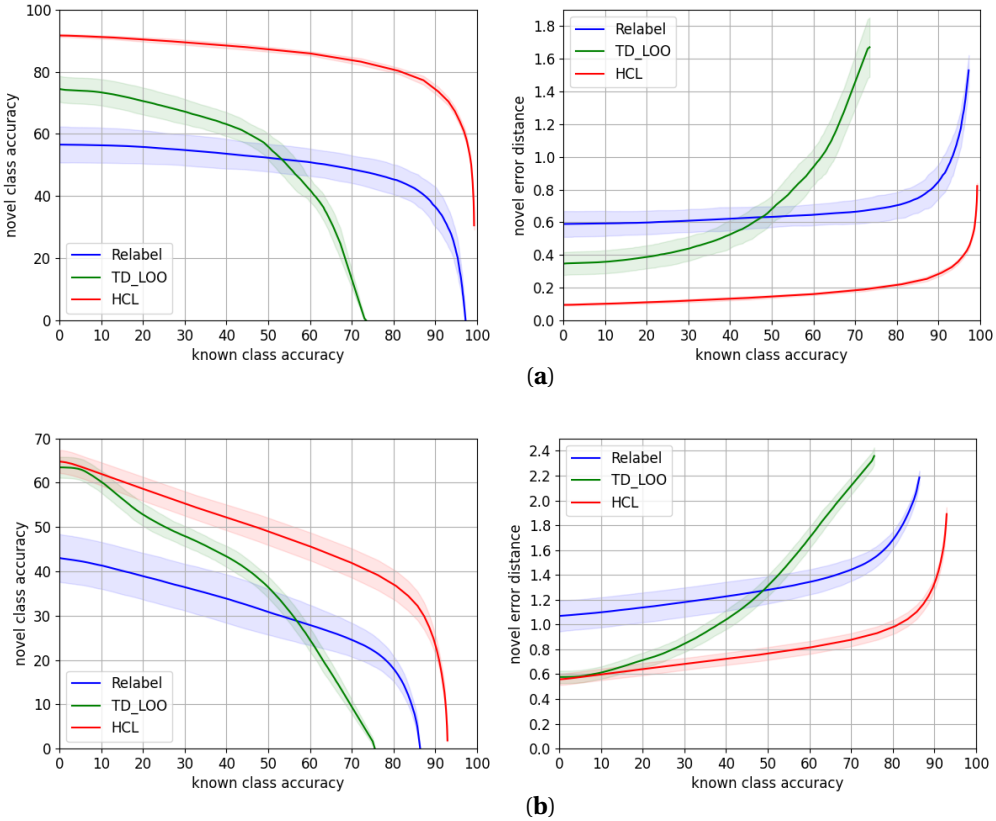


Figure 3.6: Novel/known accuracy trade-off and novel average hierarchical error distance over known accuracy, for HCL (red) and the state-of-the-art models TD+LOO (green) and Relabel (blue) for (**a**) TT100K and (**b**) MTSD.

We compare in Table 3.3 HCL to *TD+LOO* and *Relabel*. The reported metrics

are the average value $\pm 2\sigma$ from the set of 50 experiments. Figure 3.6 shows the novel/known accuracy trade-off and the average hierarchical error distance on the novel split over the known split accuracy.

On both datasets, HCL consistently outperforms *Relabel* [47] and *TD+LOO* [47] by a large margin throughout all the ranges of accuracy both in terms of accuracy and average novel hierarchical error distance. Our results suggest our approach is more suitable for traffic signs datasets. A possible explanation is related to the taxonomy of these datasets. Both *TD+LOO* and *Relabel* solely rely on the cross-entropy loss, but HCL learns an embedding of discriminative features that could benefit from taxonomies related to the visual appearance of the classes, *e.g. prohibition* signs have common visual features, while *carnivore* images have not an indistinguishable visual feature.

3.6.2 Training Strategies

We consider three different settings to train HCL. In particular, we train HCL on top of features that are extracted from ResNet-101 models that are previously trained, either on only ImageNet, or finetuned to the target dataset via the cross-entropy loss. The third setting we compare is when we train simultaneously HCL and the ResNet-101 backbone, that is pretrained on ImageNet. We keep the fine-tuning procedure that is detailed in the previous Section 3.6.1.

Table 3.4: Comparison of different training strategies for HCL on MTSD and TT100K. I stands for ImageNet features, F for finetuned features and B for training both the backbone and HCL simultaneously. Performance is measured by the novel/known accuracy AUC and the novel accuracy and average hierarchical error distance $\bar{d}_h$ at 50%, 70% and 80% known accuracy points.

	AUC	Novel acc $\pm 2\sigma \uparrow$			Novel $\bar{d}_h$ $\pm 2\sigma \downarrow$		
	$\pm 2\sigma$	@50%	@70%	@80%	@50%	@70%	@80%
				TT100K			
I	54.0±3.2	63.2±2.8	45.3±4.5	27.4±4.6	0.43±0.03	0.69±0.02	0.95±0.02
F	**84.1±0.7**	**87.2±0.8**	**83.7±1.1**	**80.7±1.0**	**0.15±0.01**	**0.18±0.01**	**0.22±0.01**
B	71.4±4.1	80.0±7.8	60.6±9.3	50.9±6.5	0.22±0.08	0.42±0.09	0.54±0.06
				MTSD			
I	25.9±1.0	34.3±1.4	00.0±0.0	00.0±0.0	1.36±0.02	-	-
F	**44.2±1.6**	**47.7±2.1**	**40.5±2.3**	**35.8±2.2**	**0.78±0.04**	**0.89±0.05**	**0.99±0.05**
B	43.1±8.2	47.4±11.3	36.7±10.9	30.8±7.1	1.01±0.20	1.16±0.22	1.26±0.19

Using the hyperparameters from Table A.1, we repeat each new experiment for 10 times, as some training variants we compare in this section are time consuming

and HCL was shown to be not so variable in previous results.

Table 3.4 reports the average metrics $\pm 2\sigma$ for these three training strategies at 50%, 70% and 80% known accuracy points, while the performance throughout the entire range is depicted at Figure 3.7.

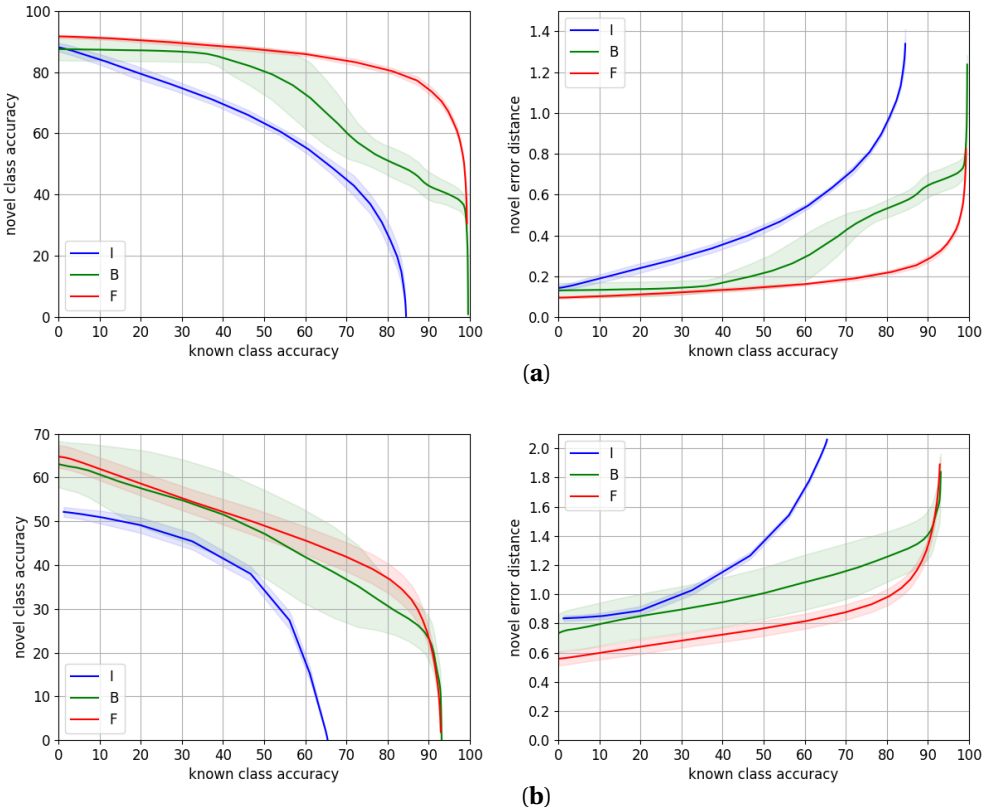


Figure 3.7: Novel/known accuracy trade-off and novel average hierarchical error distance over known accuracy, for different training strategies for HCL on (**a**) TT100K and (**b**) MTSD. I stands for ImageNet features, F for finetuned features and B for training both the backbone and HCL simultaneously.

The gap of performance on both datasets between using features from a network only trained on Imagenet, and features finetuned to the target dataset, justifies the

need of performing such fine-tuning. Specially because in a traffic sign recognition application we aim to maximize the novel accuracy at the highest known accuracy range.

On **TT100K**, training HCL on top of finetuned features works significantly better than training jointly HCL and the ResNet-101 backbone. This is probably because fine-tuning on TT100K is overfitting the dataset, which has few samples of a very specific kind of data (traffic signs). Then, training HCL from high quality features as a starting point is much easier than jointly learning suitable features along with proper class *prototypes* that are consistent with the taxonomy.

However, **MTSD** is a much larger dataset with a greater number of classes with higher inter and intra-class variability, as discussed in section 3.4.2. The gap of performance in this case is therefore smaller. The accuracy curve overlaps for almost the entire range, although the gap in error distance is consistent. This means for a very similar number of correct predictions, the errors of the wrong predictions are smaller when we use finetuned features, presumably because these finetuned features allow to do a more precise classification of novel samples. The reason might be learning two objectives, *i.e.* class *prototypes* and suitable features, is a more challenging task than only learning the *prototypes* with a fixed set of features. This involves a less noisy signal to learn from, since features are not being updated during training. This effect, that also occurred on TT100K at a smaller scale, is magnified with larger datasets with high intra-class variability, as in this case.

As expected, the variability of the experiments when we train the backbone is higher than when we only train HCL, for both datasets. This is due to the additional variability introduced by training ResNet-101.

To conclude, our proposed approach, HCL, reaches its highest performance when trained from features finetuned to the target dataset. It is able to predict correctly 75% and 24% of novel samples, for TT100K and MTSD respectively, when we predict known samples with 90% of accuracy. It also predicts novel samples with an accuracy of 81% and 36% at 80% known accuracy for TT100K and MTSD, respectively.

3.6.3 Ablation Study of Hierarchical Cosine Loss

In order to analyze the individual contribution of the terms of HCL (Eq. 3.1), we conduct the following ablation study. We take as a baseline the contribution of only the Normalized Softmax loss (NSL) L_{NS}, then adding the contribution of the remaining terms, that will be finally compared to an experiment in which all the terms contribute to the training. The latter is the best performing HCL experiment, according to the validation set. We train the terms of HCL over a set of constant features, finetuned to the target dataset. These finetuned features correspond to

those used in the previous experiments. Note this independent training of HCL allows to isolate the effect of the loss. Otherwise, training jointly the backbone and HCL would introduce a variability that would mask the actual variation of the individual loss terms.

In Table 3.5 we assess the individual performance of the different terms of HCL, defined in section 3.3.2, on features finetuned to MTSD or TT100K. For each dataset, the first row shows, as a baseline, the metrics when we set the HCL regularization parameters to $\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\} = \{1, 0, 0, 0\}$, *i.e.* we train using only the NSL L_{NS}. In the experiments of the next rows we keep $\lambda_{NS} = 1$ and add the different terms on each experiment, *e.g.* the second row corresponds to $\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\} = \{1, 10, 0, 0\}$ where only the NSL L_{NS} and Hierarchical Centers term L_{HC} are contributing to the training. Similarly, the third row corresponds to the experiments where we use only the L_{NS} and L_{CT} terms with regularization parameters $\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\} = \{1, 0, 1, 0\}$, and in the fourth row we train using L_{NS} and L_{HT} regularized by $\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\} = \{1, 0, 0, 0.1\}$. The last row, where $\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\} = \{1, 10, 1, 0.1\}$, reports the performance of the full version of HCL, including all the terms. Despite the NSL weight λ_{NS} is always set to 1, we made sure its contribution to the loss was not leading the training, *i.e.* the loss that is being analyzed individually at each case, is not being neglected and actually contributes to the training. This is, we made sure the weights applied to the individual terms were appropriate to show the individual effect of the loss terms. Each training variant has been repeated for 10 times and we report an error of 2σ on Table 3.5.

For **TT100K**, the first experiment in which we only use the NSL L_{NS}, gets the best average metrics among the compared variants. However, the difference of performance is very small. In fact, if we take into account the variability of the experiments, we could consider all the variants to perform similarly. The cause of this result might be the performance on this dataset is so good that reaches a limit that is hard to surpass. Making small modifications on the loss is not translated into a significant change on performance. In this scenario, making even very small improvements is not straightforward and it would probably mean it is overfitting the dataset.

The results on **MTSD** are more enlightening, it is a more challenging dataset, closer to a real life scenario. Using the different terms of HCL always improves the average novel accuracy at 70% and 80% known accuracies w.r.t. the NSL baseline. The variant that performs best on these metrics is the full version of HCL. The distance error is also decreased for all the variants except for L_{NS}, L_{CT}, that gets equivalent performance. The best error distance is achieved by L_{NS}, L_{HT}, but as before, if we consider the variability of the results, the differences w.r.t. the full version of HCL are not significant.

A remarkable outcome we can draw from these experiments is that the Hierar-

Table 3.5: Ablation study of the HCL terms. Performance is measured by the novel/known accuracy AUC and the novel accuracy and average hierarchical error distance $\bar{d}_h$ at 70% and 80% known accuracy points. The metrics are the average of 10 experiments $\pm 2\sigma$.

Losses	$\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\}$	AUC $\pm 2\sigma$	Novel acc $\pm 2\sigma \uparrow$ @70%	Novel acc $\pm 2\sigma \uparrow$ @80%	Novel $\bar{d}_h$ $\pm 2\sigma \downarrow$ @70%	Novel $\bar{d}_h$ $\pm 2\sigma \downarrow$ @80%
				TT100K		
L_{NS}	{1,0,0,0}	**84.1±0.6**	**83.9±0.8**	**80.9±0.7**	**0.18±0.01**	**0.21±0.01**
L_{NS}, L_{HC}	{1,10,0,0}	84.0±0.4	83.7±0.5	80.7±0.7	0.19±0.01	0.22±0.01
L_{NS}, L_{CT}	{1,0,1,0}	83.8±0.9	83.6±1.3	80.6±1.2	0.19±0.01	0.22±0.01
L_{NS}, L_{HT}	{1,0,0,0.1}	**84.1±0.7**	83.8±0.9	80.7±0.8	**0.18±0.01**	0.22±0.01
HCL	{1,10,1,0.1}	**84.1±0.7**	83.7±1.1	80.7±1.0	**0.18±0.01**	0.22±0.01
				MTSD		
L_{NS}	{1,0,0,0}	41.8±0.5	37.6±0.7	33.7±0.7	0.93±0.01	1.03±0.01
L_{NS}, L_{HC}	{1,10,0,0}	42.8±1.1	38.9±1.4	34.7±1.4	0.91±0.03	1.01±0.02
L_{NS}, L_{CT}	{1,0,1,0}	42.5±2.2	38.5±2.8	34.1±2.8	0.94±0.07	1.04±0.07
L_{NS}, L_{HT}	{1,0,0,0.1}	43.7±1.2	39.9±1.7	35.7±1.8	**0.88±0.03**	**0.97±0.03**
HCL	{1,10,1,0.1}	**44.2±1.6**	**40.5±2.3**	**35.8±2.2**	0.89±0.05	0.99±0.05

chical Triplets term L_{HT} improves the average metrics at the cost of increasing the variability of the method. This is expected as this constraint introduces different information that depends on the training data. As discussed in section 3.3.2, we make triplets from the batch that is fed to the network. If batches are different, so will be the triplets. In the case of MTSD, which is a much larger dataset than TT100K, it is possible to make a much larger number of triplets, that introduce different information, consequently affecting the training result. This is also applied to the C-triplet term L_{CT} for the same reason. There are more available pairs of different classes in a larger dataset, therefore affecting the training outcome.

It is also worth to mention that the variability of L_{HC} is expected to be low due to the kind of experiments we carry. Its cost depends on the angle between class *prototypes*. Using fixed pre-computed features that are not changing throughout the training only requires to find the class *prototypes*. Training also the ResNet-101 backbone would be translated into a higher variability for this term.

In summary, this ablation study shows the proposed HCL terms can help improving the performance, as shown on MTSD results. On TT100K, they do not improve the performance of the NSL alone in average, because it already reaches a very high value. Some of the HCL terms (L_{HT}, L_{CT}) have shown to increase the potential performance at the cost of increasing the variability of the results. Triplet mining strategies might help to mitigate this issue.

3.7 Conclusions

We have addressed the problem of hierarchical novelty detection, specifically focused on traffic sign recognition. It involves classification along with detection of novel classes, and consists in predicting not only that a sample belongs to a novel class (never seen during training), but also its closest position in a semantic hierarchy of known classes. We have introduced a novel loss function, Hierarchical Cosine Loss, that learns jointly an embedding of discriminative features consistent with the class taxonomy, as well as *prototype* representations for both leaf and parent classes. HCL achieves equivalent results to state-of-the-art approaches on natural images benchmarks, AWA2 and CUB, and significantly outperforms them on traffic sign datasets. For the latter experiments, we have contributed taxonomies and corresponding training splits for TT100K and MTSD, two challenging large scale traffic signs benchmarks that simulate real data of a traffic sign recognition application. Our approach is able to detect novel samples from unknown classes at the correct nodes of the hierarchy with 75% and 24% of accuracy when we classify known classes with 90% of accuracy, for TT100K and MTSD, respectively. It also reaches 81% and 36% of novel accuracy at 80% known accuracy, for TT100K and MTSD, respectively. Finally, we have contributed an ablation study that analyzes the individual performance of the HCL terms.

4 Weakly Supervised Multi-Object Tracking and Segmentation

4.1 Introduction

Computer vision based applications often involve solving many tasks simultaneously. For instance, in a real-life autonomous driving system, tasks regarding perception and scene understanding comprise the problems of detection, tracking, semantic segmentation, etc. In the literature, however, these are usually approached as independent problems. This is the case of multi-object tracking and instance segmentation, which are usually evaluated as disjoint tasks on separate benchmarks. The problem of Multi-Object Tracking and Segmentation (MOTS) was recently defined in [97]. As an extension of the Multi-Object Tracking problem to also comprise instance segmentation, it consists in detecting, classifying, tracking and predicting pixel-wise masks for the object instances present along a video sequence.

Due to the lack of suitable datasets, the first two MOTS benchmarks were introduced in [97] in order to assess their model, which were annotated manually. The annotation procedure involves providing bounding boxes and accurate pixel-level segmentation masks for each object instance of predefined classes, plus an unique identity instance tag, consistent along the video sequence. Moreover, this needs to be done on a significant amount of data to effectively train a MOTS model. This results in a high annotation cost and makes infeasible to perform it manually. This issue can be mitigated by investigating approaches that do not require all this data to solve the MOTS task. In this chapter, we address this unexplored line of research.

We define the weakly supervised MOTS problem as the combination of weakly supervised instance segmentation and multi-object tracking. It aims at detecting, classifying, tracking and generating pixel-wise accurate masks, without providing any kind of instance segmentation annotation, the most expensive annotation type of MOTS datasets. We propose an approach that solves this task by using only detection and tracking annotations: bounding boxes along with their corresponding classes and identities. By taking advantage of multi-task learning, we design a synergistic training scheme where the supervised tasks support the unsupervised one. We are able to solve the instance segmentation task by relying on the learning

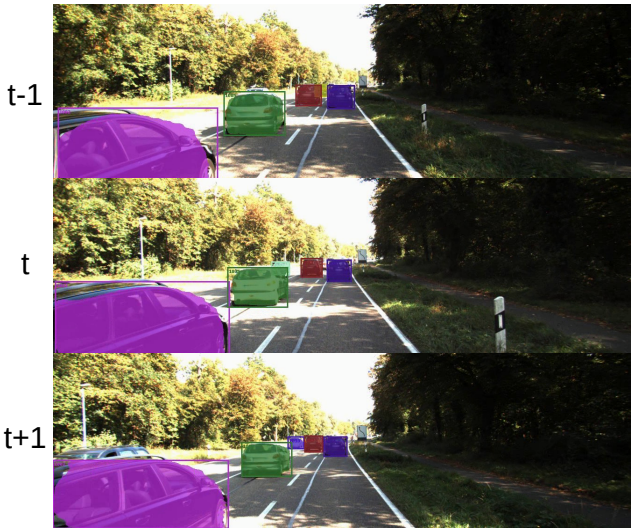


Figure 4.1: Output of our weakly supervised approach on KITTI MOTS. Different colors represent the different identities.

of the parallel supervised tasks (see Figure 4.1 for an output example). Specifically, we provide weak supervision from the classification and tracking tasks, along with RGB image level information. The learning of the instance segmentation task solely depends on this novel supervision. The proposed weak supervision consists of three losses that integrate: localization information via activation heatmaps extracted from the classification task, tracking information and RGB image level information, to refine the prediction at the objects boundaries. To the best of our knowledge, we solve for the first time the MOTS problem under a weakly supervised setting.

Our main contributions are the following:

- We define the weakly supervised MOTS problem as joint weakly supervised instance segmentation and multi-object tracking. This is the first work that, to the best of our knowledge, considers this variant of the MOTS problem and solves it not using any kind of instance segmentation annotations.

- We design a novel training strategy to address weakly supervised MOTS. The different branches of our architecture, MaskR-CNN based, act synergistically to supervise the instance segmentation task, *i.e.* classification and tracking actively help segmentation.

- We compare our method to the fully supervised baseline on the KITTI MOTS dataset, showing that the drop of performance, on the MOTSP metric is just 12% and 12.7% for cars and pedestrians, respectively.

- Finally, we provide an ablation study about the contribution of the components of our approach.

4.2 Related Work

4.2.1 Multi-Object Tracking and Segmentation

The MOTS problem was introduced in [97]. The solution proposed by the authors consists in a MaskR-CNN based architecture that comprises an additional tracking branch that learns an embedding, later used to match the object instances along the frame sequence. Despite it is a recently introduced topic, there already exist works related to the MOTS problem on a fully-supervised setting. In [39], instead of joining the problems of instance segmentation and tracking, they solve jointly panoptic segmentation and tracking. A similar idea to our approach, in the sense of using multi-object tracking to help other tasks, is presented in [63]. On their approach, MOTSFusion, tracking helps 3D reconstruction and vice-versa. Very recently, a new framework has been proposed in [110] along with a new MOTS dataset, APOLLO MOTS. Differently from the previous works, the instance segmentation task is not solved in a two stage manner from the bounding box predictions. Instead, they use the SpatialEmbedding method, which is bounding box independent and faster. An extension is done in [111].

There are no previous works addressing weakly supervised settings of the MOTS problem. However, stressing the importance of the need of annotations for MOTS, an automatic annotation procedure for MOTS benchmarks was proposed in [74], where the authors also presented a similar architecture to [97]. As the result of their automatic annotation pipeline, they obtain instance segmentation masks and tracking annotations. However, the masks are obtained from a network that is previously trained using instance segmentation masks from a different benchmark, with a domain gap presumably small with respect to the target dataset. Our model instead, is trained with no previous knowledge of how a mask "looks like".

4.2.2 Weakly Supervised Segmentation

The literature in the field of semantic segmentation is extensive and there exist many works that address the weakly supervised setting. A widely used strategy is to predict an initial weak estimate of the mask, that is then refined by using extra

information extracted from the image, *e.g.* using Conditional Random Fields (CRF) as a post-processing step is a common approach to get precise boundaries of the objects.

Some works that follow such strategy are [43, 83], which employ a dense CRF [44] to improve their mask prediction. In [43], the authors propose to minimize the KL divergence between the outputs of the network and the outputs of the CRF, while in [83], they smooth their initial mask approximation by using the CRF. They then minimize a loss that computes the difference between the network prediction and the CRF output. Both of them use activations of the network as an initial mask estimation. More recently, [90] employs CRF post-processing to refine initial rectangle-shaped proposals, that are later used to compute the mean filling rates of each class. With their proposed filling rate guided loss, they rank the values of the score map, then selecting the most confident locations for back propagation and ignoring the weak ones.

The mean-field inference of the CRF model [44] was later formulated in [125] as a Recurrent Neural Network, which allows to integrate it as a part of a CNN, and train it end-to-end. This formulation is used in the architecture from [4, 53]. In [4], it is used to refine the initial semantic segmentation and the final instance segmentation predictions. A weakly supervised panoptic segmentation method is proposed in [53]. Two outputs are proposed as the initial masks. If bounding box annotations are available, they use a classical foreground segmentation method. Otherwise, the approximate masks are localization heatmaps from multi-class classification [86], similarly to us. However, their classification network is previously trained and only used to extract the heatmaps. We instead, train all the classification, detection, instance segmentation and tracking tasks simultaneously. Also, we do not have an independent classification network dedicated to extract the heatmaps, it is part of the main architecture. Another advantage of our method is that it extracts the heatmap individually for each ROI proposal, instead of doing it for the whole image.

Differently from the previous methods, the work of [95], that considers the problem of training from partial ground truth, integrates the CRF regularizer into the loss function, then avoiding extra CRF inference steps. Their weakly-supervised segmentation loss function is composed by a ground truth plus a regularization term. They propose and evaluate several regularization losses, based on Potts/CRF, normalized cut and KernelCut regularizers.

4.2.3 Video Object Segmentation

Video Object Segmentation (VOS) is a problem related to ours, as it also comprises tracking and segmentation. In VOS, all the salient objects that appear in the sequence must be segmented and tracked, regardless of their category. Salient objects

are those that catch and maintain the gaze of a viewer across the video sequence. Differently, in MOTS, we only track and segment objects that belong to specific classes of interest, therefore needing a classification model. Some recent works in the field of VOS are [21, 93, 118]. If we add classification to VOS, then distinguishing object instances, it becomes Video Instance Segmentation (VIS) [8, 57, 112]. The datasets designed to assess this task do not usually present strong multi-object interaction, then lacking hard scenarios with occlusions and objects that disappear and enter again to the scene, as it is characteristic of MOTS benchmarks.

There exist semi and unsupervised approaches of the VOS problem. In the semi-supervised setting the masks of the objects to be tracked are given in the first frame. Only these objects need to be tracked and segmented throughout the rest of the video. The unsupervised approach, however, consists in detecting all the possible objects in the video and track and segment them throughout the whole sequence. The work of [64] addresses the unsupervised VOS problem with a MaskR-CNN based architecture, trained on COCO. They do the inference for the 80 classes of COCO, using for mask prediction a very low (0.1) confidence threshold, then merging the mask predicted for all the categories, taking the most confident one when there is overlapping. This method was extended to VIS by just adding classification, also provided by Mask R-CNN.

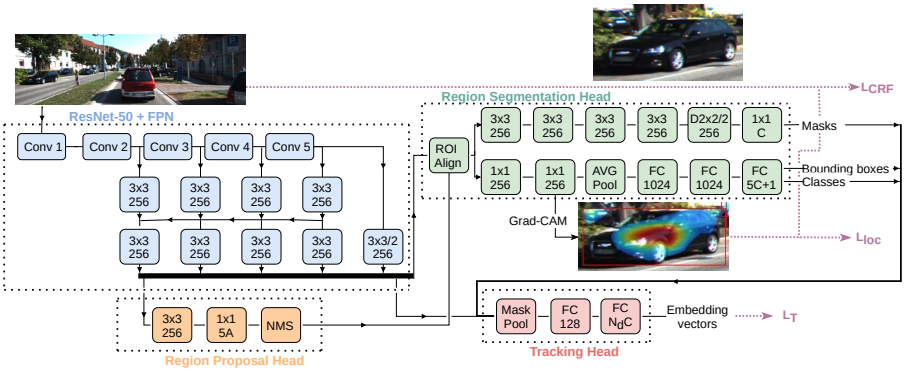


Figure 4.2: Overview of our architecture. We modify MOTSNet [74] by adding 1×1 convolutional layers on the classification and detection branch to extract localization information via Grad-CAM [86] heatmaps. We show in purple the losses, L_{loc}, L_{CRF} and L_T, that supervise the instance segmentation task in the weakly supervised setting.

4.3 Method

We build upon the MOTSNet architecture proposed in [74]. It is a MaskR-CNN based architecture with an additional Tracking Head. Its backbone is composed by a ResNet-50 followed by a Feature Pyramid Network which extracts features at different resolutions, later fed to a Region Proposal Head (RPH). The features of the bounding box candidates predicted by the RPH enter the Region Segmentation Head, that learns the classification, detection and instance segmentation tasks and the Tracking Head, that learns an embedding. We add two 1×1 convolutional layers at the classification and detection branch of the Region Segmentation Head, aimed at Grad-CAM [86] computation for the ROI proposals, as described in section 4.3.1. This is needed to extract activation information, as the original branch does not include any convolutional layer. The complete architecture is shown in Figure 4.2.

First, we describe the general fully supervised setting to finally introduce our weakly supervised approach. To train the model under a fully supervised setting, we employ the loss function defined in [74], with minor differences in the tracking loss, described below. The loss function L is then defined as

$$L = L_T + \lambda(L_{RPH} + L_{RSH}), \tag{4.1}$$

where L_T, L_{RPH} and L_{RSH} denote the Tracking, Region Proposal Head and Region Segmentation Head losses, respectively. We refer the reader to [73] for a detailed description of the two latter.

Tracking. MOTSNet is based on MaskR-CNN but comprises a new Tracking Head (TH) that learns an embedding at training time and predicts class specific embedding vectors for each proposal. The TH first applies the *mask-pooling* [74] operation on the input features, thereby only considering the foreground of the proposal to compute its embedding vector. This embedding is trained by minimizing a hard-triplet loss [35], so that instances of the same object are pushed together in the embedding space, while instances of different objects are pushed away. The distance in the embedding space is then used at inference time to associate the proposals and build the tracks. We define the distance as the Cosine distance $d(v,w) = \frac{v\cdot w}{\|v\|\|w\|}$ between two embedding vectors v and w.

Then, the tracking loss L_T is defined as

$$L_T = \frac{1}{|\check{\mathscr{R}}|}\sum_{\check{r}\in\check{\mathscr{R}}} \max\left(\max_{\hat{r}\in\hat{\mathscr{R}}|id_{\hat{r}}=id_{\check{r}}} d(a_{\hat{r}},a_{\check{r}}) - \min_{\hat{r}\in\hat{\mathscr{R}}|id_{\hat{r}}\neq id_{\check{r}}} d(a_{\hat{r}},a_{\check{r}}) + \alpha, 0\right), \tag{4.2}$$

where $\check{\mathscr{R}}$ denotes the set of positive matched region proposals in the batch. The positive proposals are those that match a bounding box from the ground truth

with an IoU > 0.5. $a_{\check{r}}$ and $id_{\check{r}}$ stand for the corresponding embedding vector and assigned identity from the ground truth track, of the proposal $\check{r} \in \tilde{\mathscr{R}}$. α is the margin parameter of the hard triplet loss.

At inference time, the tracking association is performed as follows. To link positive proposals from consecutive frames, we first discard those whose detection confidence is lower than a threshold. We then compute a similarity function for each pair of objects. We consider the pairs between the current frame objects and the objects present in the previous frames comprised in a temporal window whose length is previously decided.

The similarity function $Sim(\check{r}, \hat{r})$ of two proposals $\check{r}$ and $\hat{r}$ takes into account the embedding distance and the bounding box overlapping as

$$Sim(\check{r}, \hat{r}) = IoU(b_{\check{r}}, b_{\hat{r}}) d(a_{\check{r}}, a_{\hat{r}}), \tag{4.3}$$

where $b_{\check{r}}$, $b_{\hat{r}}$ are the predicted bounding boxes associated to $\check{r}$ and $\hat{r}$, respectively. From this similarity, we define a cost

$$Cost(\check{r}, \hat{r}) = \left[\max_{\check{r}, \hat{r} \in \tilde{\mathscr{R}}} Sim(\check{r}, \hat{r}) \right] - Sim(\check{r}, \hat{r}). \tag{4.4}$$

Finally, the matching is solved by using the Hungarian algorithm.

4.3.1 Weakly supervised approach

The loss function that trains the model under a fully supervised setting is defined in Eq. 4.1, where L_{RSH} is

$$L_{RSH} = L_{RSH}^{cls} + L_{RSH}^{bb} + L_{RSH}^{msk}, \tag{4.5}$$

L_{RSH}^{cls}, L_{RSH}^{bb} and L_{RSH}^{msk} stand for the classification, bounding box regression and mask segmentation losses of the Region Segmentation Head. In the fully supervised case, L_{RSH}^{msk} corresponds to a cross-entropy loss that compares the instance segmentation ground truth to the predicted masks.

In our weakly supervised setting, we do not have any instance segmentation ground truth available. To train the instance segmentation task, we propose a new approach that benefits from the multi-task design of the MaskR-CNN base architecture, *i.e.* it has a common backbone followed by task-specific heads. We exploit this architecture so that the different branches of MOTSNet act in a synergistic manner, guiding the unsupervised task. In particular, we propose a new definition of L_{RSH}^{msk},

$$L_{RSH}^{msk} = L_{loc} + \lambda_{CRF} L_{CRF}, \tag{4.6}$$

where L_{loc} and L_{CRF} stand for the Foreground localization and CRF losses, respectively and λ_{CRF} is a regularization parameter.

Foreground localization loss L_{loc}. To provide information to the network about where the foreground is, we use a localization mechanism. In particular, we propose Grad-CAM [86], *i.e.* weak localization heatmaps obtained from the activations and gradients that flow trough the last convolutional layer of a classification network, when it classifies the input as a certain class. Since our architecture naturally comprises a classification branch, we take advantage of that, using the MOTSNet classification branch to compute Grad-CAM heatmaps. As explained in section 4.3, we add two 1×1 convolutional layers to the classification and detection branch, before the fully connected layers. The Grad-CAM heatmaps are computed then on the second added convolutional layer by using the implementation variant discussed in section 4.3.2.

Let $\mathscr{R}$ be the set of bounding boxes from the ground truth. For every bounding box $r \in \mathscr{R}$, we compute the Grad-CAM heatmap G^r corresponding to that ground truth region, for its associated class. We normalize it so that $G^r \in [0,1]^{28\times 28}$. The heatmaps G^r are intended to produce mask pseudo labels to learn from. For a region proposal $\check{r}$, its corresponding pseudo label $Y^{\check{r}} \in \{0,1,\emptyset\}^{28\times 28}$ is a binary mask generated from the heatmaps, where $\emptyset$ denotes a void pixel that does not contribute to the loss. The assignment of the pseudo label $Y^{\check{r}}_{ij}$ to the cell (i,j) is defined as

$$Y^{\check{r}}_{ij} = \begin{cases} 0 & \forall\, i\,j \notin \mathscr{P}^r \;\; \forall r \in \mathscr{R} \\ 1 & \text{if } G^r_{ij} \geq \mu_A \;\; \forall\, i\,j \in \mathscr{P}^r \\ \emptyset & \text{if } G^r_{ij} < \mu_A \;\; \forall\, i\,j \in \mathscr{P}^r\,, \end{cases} \tag{4.7}$$

where $\mathscr{P}^r$ is the set of pixels that belong to the area defined by the ground truth bounding box r. We consider as foreground the pixels of the ground truth bounding boxes whose Grad-CAM value G^r is above a certain threshold μ_A and background all the pixels outside the bounding boxes. We ignore those pixels that are inside the bounding boxes but below the threshold. Figure 4.3 shows a visualization exemple of the generated pseudo labels.

Then, the foreground localization loss L_{loc} is a cross entropy loss, defined for a proposal $\check{r}$ as

$$L_{loc}(Y^{\check{r}}, S^{\check{r}}) = -\frac{1}{|\mathscr{P}^{\check{r}}_Y|} \sum_{(i,j)\in\mathscr{P}^{\check{r}}_Y} Y^{\check{r}}_{ij} log S^{\check{r}}_{ij} - \frac{1}{|\mathscr{P}^{\check{r}}_Y|} \sum_{(i,j)\in\mathscr{P}^{\check{r}}_Y} (1 - Y^{\check{r}}_{ij}) log(1 - S^{\check{r}}_{ij})\,, \tag{4.8}$$

where $S^{\check{r}} \in [0,1]^{28\times 28}$ denotes the mask prediction for the proposal $\check{r}$ for its predicted

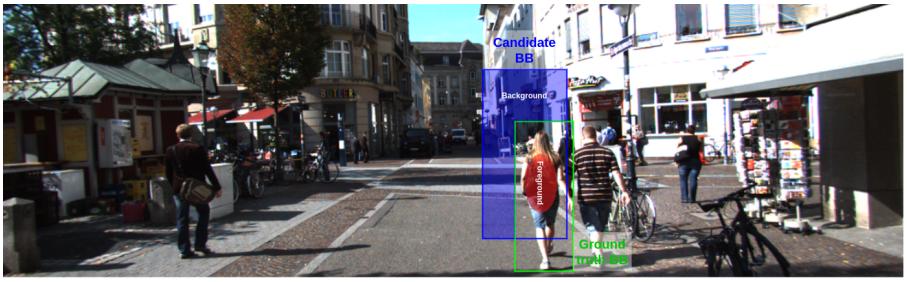


Figure 4.3: Visualization of the generated pseudo labels. The blue and green boxes represent a possible candidate and a ground truth bounding boxes, respectively. The blue shaded area will be considered as background, while the red shaded area (heatmap pixels above the threshold) will be considered foreground in the Foreground localization loss.

class, whose entries $S^{\check{r}}_{ij}$ are the probability of cell (i, j) to belong to the predicted class. $\mathscr{P}^{\check{r}}_{Y} \subset \mathscr{P}^{\check{r}}$ denotes the set of all the non-void pixels in the 28×28 pseudo label mask $Y^{\check{r}}$, letting $\mathscr{P}^{\check{r}}$ be the set of all the pixels in $Y^{\check{r}}$. The loss values of all the positive proposals (those with a bounding box IoU > 0.5) are averaged by the number of proposals to compute the loss.

CRF Loss L_{CRF} . We use the loss proposed in [95] to improve the instance segmentation prediction on the object boundaries. This loss integrates CRF regularizers, that can act over a partial input, improving the quality of the predicted mask. Thus, we avoid additional CRF inference steps that many weakly supervised segmentation methods do [4, 43, 53, 83]. The CRF loss L_{CRF} is a regularization loss, result of applying a relaxation of the dense CRF regularizer.

The Potts model can be expressed as

$$\sum_{(ij,kl)\in\mathscr{P}^{\check{r}}} W_{ijkl}[S^{\check{r}}_{ij} \neq S^{\check{r}}_{kl}] \approx \sum_{k} S^{\check{r}k'} W(1 - S^{\check{r}k}) = L_{CRF}(S^{\check{r}}), \tag{4.9}$$

The right hand side of the equation above is a quadratic relaxation of the Potts model, proposed by [95]. L_{CRF} provides the cost of a cut between segments, as the Potts model in the left hand side of Eq. 4.9. W represents an *affinity matrix*, *i.e.* the matrix of pairwise discontinuity costs, k denotes the class and $S^{\check{r}k} \in [0,1]^{128\times128}$ is the predicted mask for that class, resized from 28×28 to 128×128 in order to extract quality information from the RGB image. Following the implementation of [95], we consider a dense Gaussian kernel over RGBXY, then W is a relaxation of

DenseCRF [45]. The loss is implemented by computing its gradient as

$$\frac{\partial L_{CRF}(S^{\check{r}})}{\partial S^{\check{r}k}} = -2WS^{\check{r}k} \tag{4.10}$$

The gradient computation becomes standard Bilateral filtering that can be implemented by using fast methods such as [1]. Similarly as with the L_{loc} loss, we average the losses for all the positive proposals.

Tracking loss L_T. As described before, the TH first applies the *mask pooling* operation, *i.e.* the embedding vector predicted by the TH only considers the foreground according to the predicted mask. The tracking loss is then also indirectly supervising the instance segmentation branch.

In summary, the training of the instance segmentation branch is guided by the linear combination of these losses. The algorithm overview is depicted in Figure 4.2. The RGB image is used along with the mask prediction to compute L_{CRF}, while the ground truth bounding boxes are used to compute Grad-CAM heatmaps that produce pseudo labels to learn from, via a cross-entropy loss applied on the mask prediction. Finally, the TH employs the mask prediction to produce embedding vectors, then indirectly supervising the instance segmentation task. The effect of the combination of the aforementioned losses is shown on Figure 4.4, where we show the initial Grad-CAM heatmaps that are used to produce pseudo labels and the final predicted mask by the weakly supervised mask branch.

4.3.2 Grad-CAM analysis

In the original implementation of [86], the Grad-CAM heatmap $G^c \in \mathbb{R}^{28x28}$ for a certain class c is computed as

$$G^c = ReLU\left(\sum_k \alpha_k^c A^k\right), \tag{4.11}$$

where the importance weights α_k^c are defined as the global-average-pooled gradients $\frac{\partial y^c}{\partial A_{ij}^k}$ over the width and height dimensions i, j,

$$\alpha_k^c = \underbrace{\frac{1}{Z}\sum_i \sum_j}_{\text{global average pooling}} \frac{\partial y^c}{\partial A_{ij}^k}, \tag{4.12}$$

where y^c is the classification score for class c and A^k are the activations of the feature map k of the last convolutional layer in the classification architecture.

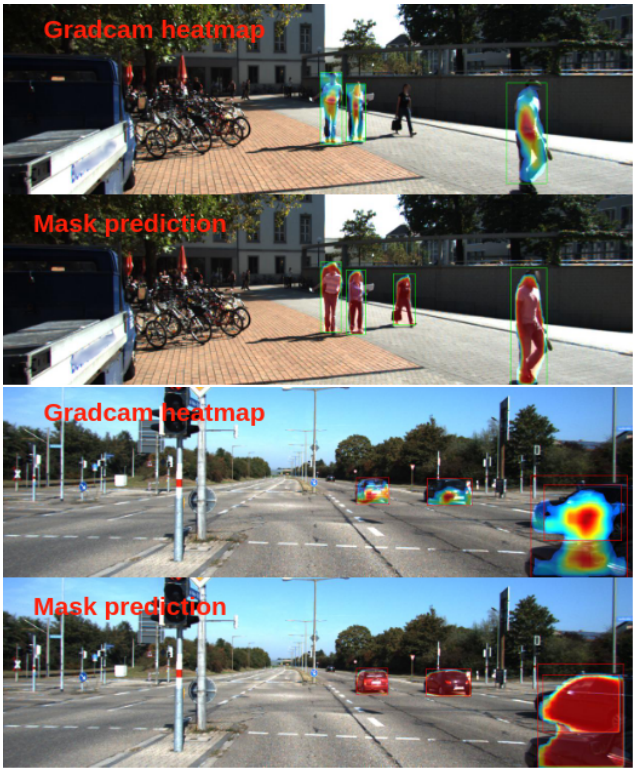


Figure 4.4: Pairs of Grad-CAM heatmaps used as a cue and the corresponding predicted masks.

We instead, use the absolute value of α_k^c in our implementation, then not needing the ReLU operation. The ReLU is intended to only consider the features that have a positive influence on the class of interest, as negative pixels are likely to belong to other categories, according to the authors. By using our alternative, we do not discard the weights that are big in magnitude but of negative sign, which in our experiments leaded to better instance segmentation cues. A comparison of the computed Grad-CAM heatmaps when using both the original implementation and the absolute weights variant is shown in Figure 4.5. The original Grad-CAM implementation can lead us to incomplete or not so suitable heatmaps to act as an initial

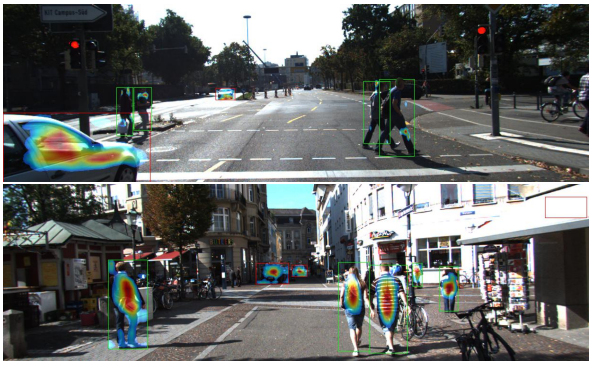


Figure 4.5: Comparison of Grad-CAM heatmaps when using the original Grad-CAM definition (top) and an implementation variant that uses the absolute value of the global-average-pooled gradients (bottom). The activations are color-coded in the heatmap from the lowest (blue) to the highest (red).

approximate of the masks. In our variant, while the highest value is located in the foreground of the object, the high activation areas cover a region of the foreground that can also be useful.

4.4 Experiments

We assess the performance of our method on the most representative MOTS benchmark, KITTI MOTS [97]. It provides balanced training and validation sets of cars and pedestrians. It is comprised of 21 sequences, extracted from the original KITTI tracking dataset, and a total of 8k frames that contain 11k pedestrian instances and 27k car instances.

4.4.1 Metrics

The MOTS performance is evaluated by the metrics defined in [97]. The authors proposed an extension of the MOT metrics [7] to assess the instance segmentation performance. Instead of considering the IoU of the predicted bounding boxes with the ground truth, as in the original metrics, they define them in terms of the mask IoU, as follows

$$sMOTSA = \frac{\widetilde{TP} - |FP| - |IDS|}{|M|} \tag{4.13}$$

$$MOTSA = \frac{|TP| - |FP| - |IDS|}{|M|} \tag{4.14}$$

$$MOTSP = \frac{\widetilde{TP}}{|TP|}, \tag{4.15}$$

where M stands for the set of ground truth masks, IDS is the number of identity switches, TP account for the masks mapped to a ground truth mask with an IoU > 0.5, $\widetilde{TP}$ is the sum of IoUs between all the predicted and ground truth masks whose IoU is at least 0.5, that is, the sum of the IoUs between the predicted masks counted as TP and their associated ground truth.

MOTSP is a pure segmentation metric; it measures the IoU of the TP predicted masks with the ground truth, which provides a measurement of the segmentation quality alone. MOTSA and sMOTSA also consider the detection and tracking performance, being sMOTSA more restrictive on the instance segmentation contribution. MOTSA only considers the number of predicted masks with an IoU > 0.5 with the ground truth, while sMOTSA counts the IoU value itself, thus penalizing low IoUs, despite being greater than 0.5.

4.4.2 Experimental setup

To show the effectiveness of our method, our backbone ResNet-50 is just pretrained on ImageNet. Pretraining on other benchmarks significantly boosts the performance of the models, as shown in [74]. However, we are not interested in optimizing a fully supervised baseline but in comparing the proposed weakly supervised approach with respect to the fully supervised baseline under the same pre-training conditions.

On our main experiments, we set the hyperparameters to the values reported in Table 4.1. Training is run on four V100 GPUs with 32GB of memory.

4.4.3 Weakly supervised approach

Since there are no previous works on weakly supervised MOTS, we compare the performance of our weakly supervised approach to the performance of our same model under the fully-supervised setting. To demonstrate that our model can achieve state-of-the-art performance under the supervised setting, we compare it against the current state of the art models under the same training conditions, *i.e.* just pre-training the ResNet-50 backbone on ImageNet. In Table 4.2, on the top section, we compare the performance of our method trained in a fully supervised manner, with the state-of-the-art model [74]. The second section shows the performance of our weakly supervised approach. Our model on both supervised

Table 4.1: Hyperparameters.

Hyperparameter	Value
Training	
Optimizer	SGD
Learning rate	0.02
Number of Epochs	150
Total batch size	24
Embedding dimensionality N_d	8
Hard triplet loss margin α	0.2
Loss weight λ_{CRF}	$2 \cdot 10^{-7}$
Grad-CAM threshold μ_A	0.5
Tracking	
Length of temporal window	10
Detection threshold	0.9

Table 4.2: Results of our approach on KITTI MOTS. The ResNet50 backbone is just pretrained on ImageNet for all the models reported.

Method	sMOTSA		MOTSA		MOTSP	
	Car	Ped	Car	Ped	Car	Ped
Fully supervised						
MOTSNet [74]	69.0	45.4	78.7	61.8	88.0	76.5
Ours	69.1	35.1	80.1	52.0	87.0	75.3
Weakly supervised						
Ours	54.6	20.3	72.5	39.7	76.6	65.7
Relative performance drop	21.0	42.2	9.5	23.7	12.0	12.7

and weakly supervised settings uses the same training hyperparameters (see Table 4.1). When our model is trained on a supervised setting, it achieves slightly superior performance than the state of the art on cars, but is inferior on some metrics for pedestrians. However, MOTSP, defined in Eq. 4.15, measures the quality of the segmentation masks without taking into account the detection or tracking performance. Our values on this metric, when we train fully supervised, are equivalent to the state of the art on both classes.

Finally, the relative drop of performance when training weakly supervised with respect to the supervised case is shown at the bottom line of the table. The performance drop on MOTSP is just a 12.0 % and 12.7 % for cars and pedestrians, respectively. This indicates the drop in segmentation quality is not drastic, considering that our model has never been trained with any mask annotation. Regarding MOTSA and sMOTSA, the performance is significantly worse on pedestrians than on cars due to the nature of pedestrians masks. Pedestrians are smaller objects and present more irregular shapes, then retrieving precisely the edges on 128×128 patches is harder. Moreover, Grad-CAM heatmaps can sometimes present high values on the surrounding area of the legs, which leads to incorrect foreground information. Qualitative results are shown on Figure 4.6.

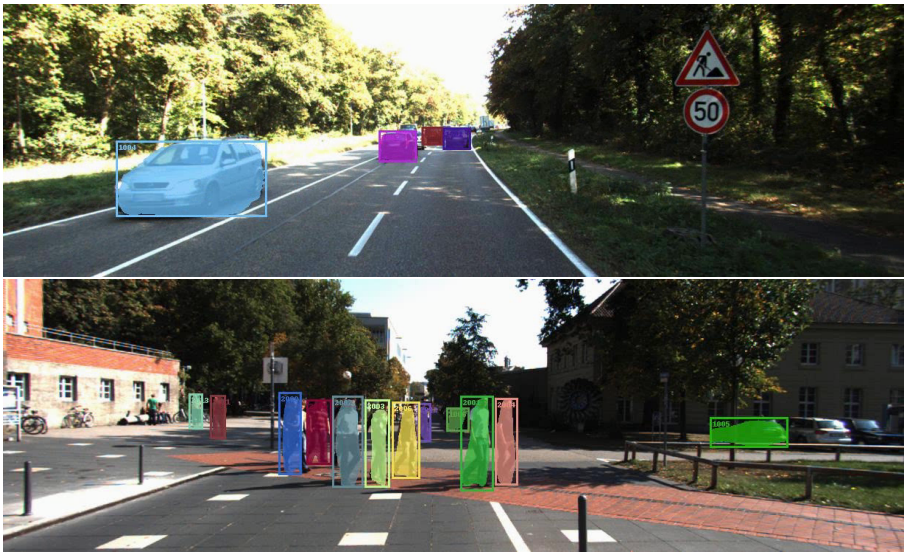


Figure 4.6: Qualitative results on test sequences of KITTI MOTS. Different colors represent the different identities.

Table 4.3: Results of the ablation study on the weakly supervised approach on KITTI MOTS (run on a previous weaker baseline).

Weakly supervised losses	sMOTSA		MOTSA		MOTSP	
	Car	Ped	Car	Ped	Car	Ped
$L_{loc} + L_{CRF} + L_T$	49.3	13.1	67.6	32.0	75.0	64.8
$L_{loc} + L_T$	44.3	10.2	66.9	30.7	69.6	63.5
$L_{loc} + L_{CRF}$	55.0	11.0	73.0	31.2	76.7	62.5

4.4.4 Ablation study

In order to assess the contribution of our proposed losses to the instance segmentation supervision, we conduct an ablation study in which we test the overall performance when removing the supervision of each loss individually. In the case of the L_T loss, we still train the Tracking Head and consider the predicted foreground of the ROIs to compute the tracking embedding vectors, but we do not propagate the gradients to the instance segmentation branch. Thus, we still train the tracking task but it does not affect to the instance segmentation supervision.

On Table 4.3, we report the performance of our approach when training with the three losses on the first row. The ablation study was performed in a weaker baseline than our main results from Table 4.2. The second and third row correspond to the experiments, trained with the same hyperparameters, when removing the supervision of L_{CRF} and L_T losses, respectively. The L_{CRF} loss clearly helps the supervision, as all the metrics suffer a performance drop when it is not applied. The tracking loss L_T, however, does help on pedestrians but not on cars. Then, the contribution of the *mask-pooling* layer as a form of supervision on the weakly supervised case is not always positive.

4.5 Conclusions

We have introduced the problem of weakly supervised MOTS, *i.e.* the joint problem of weakly supervised instance segmentation and tracking. We have contributed a novel approach that solves it by taking advantage of the multitask problem we address. Our architecture is trained in a synergistic manner so that the supervised tasks support the learning of the unsupervised one. In particular, we extract Grad-CAM heatmaps from the classification head, which encode foreground localization

information and provide a partial foreground cue to learn from, together with RGB image level information that is employed to refine the prediction at the edges of the objects. We have evaluated our method on KITTI MOTS, the most representative MOTS benchmark, and shown that the drop of performance between the fully supervised and weakly supervised approaches on MOTSP is just a 12 and 12.7 % for cars and pedestrians, respectively. Finally, we have provided an analysis of the components of our proposed method, assessing their individual contribution.

5 Conclusions and Future work

5.1 Conclusions

In this PhD dissertation we have addressed the problem of metric learning applied to several computer vision tasks. In the following discussion we answer the research questions raised in Section 1.5.

In chapter 2, we approached person re-identification by both classical and deep learning based approaches. As classical approaches, we chose a state-of-the-art method for hand-crafted feature descriptors along with a proper metric learning algorithm, specifically designed for person re-identification. Our results show the importance of the metric learning component, noticing a dramatic performance drop when removed. However, the performance of the classical method is significantly lower than employing a naive deep learning based strategy. The latter consists in cross-entropy trained features, simply compared with Euclidean distance. This shows the superiority of deep-learning based feature description. It is able to learn discriminant features, even not needing a dedicated supervised metric-learning component. Nevertheless, we believe the performance of this approach could be further increased by training the model with the specific objective of distinguishing identities, *i.e.* training it with a metric learning objective rather than classification by cross-entropy. The main objective of this chapter was to optimize the speed of the deep learning based variant at inference time, not compromising the accuracy. To this end, we have proposed network distillation to reduce the size of the network while keeping the performance of the originally trained larger network. Our results showed that smaller networks trained by this method even outperformed the larger network. This *distilled* learning, in contrast to just learning from hard-targets, increased their generalization ability.

We addressed hierarchical novelty detection by proposing a novel loss based on metric learning in chapter 3. The current state of the art is based on probabilities instead. We showed that hierarchical taxonomies of classes can be exploited for informative novelty detection. Our loss learns class *prototypes* (*i.e.* class representations in the embedding space), that allow to assign any kind of sample (including

novel ones) to the closest known class in the embedding. This enabled to assign novel samples to their parent known classes by a distance based decision. Our model beats state-of-the art approaches on two large scale traffic sign benchmarks, Mapillary Traffic Sign Dataset (MTSD) and Tsinghua-Tencent 100K (TT100K), and performs similarly on natural images benchmarks (AWA2, CUB). For TT100K and MTSD, our approach is able to detect novel samples at the correct nodes of the hierarchy with a 81% and 36% of accuracy, respectively, at 80% known class accuracy.

Finally, chapter 4 introduced the problem of weakly supervised Multi-Object Tracking and Segmentation, *i.e.* joint weakly supervised instance segmentation and multi-object tracking, in which we do not provide any kind of mask annotation. To address it, we have designed a novel synergistic training strategy by taking advantage of multi-task learning, *i.e.* classification and tracking tasks guide the training of the unsupervised instance segmentation. We evaluated our method on KITTI MOTS, the most representative benchmark for this task, reducing the performance gap on the MOTSP metric between the fully supervised and weakly supervised approach to just 12% and 12.7 % for cars and pedestrians, respectively. Finally, we provided an analysis of the components of our proposed method, assessing their individual contribution.

5.2 Contributions

In this PhD dissertation we have researched metric learning related problems, contributing to several fields of computer vision. We have shown its importance and the need of a metric learning component in very diverse applications.

First, we have dealt with the problem of person re-identification. We have applied metric learning to compare instances and we have further studied the optimization of the trade-off between speed and accuracy of the method. The main contributions of this chapter are:

- Fast and accurate compressed person re-identification via network distillation, showing that distillation helps reducing the computational cost at inference time while even increasing the accuracy performance.
- Trade-off analysis between accuracy and computational cost at test time considering both classical and current state-of-the-art deep learning based approaches, from the perspective of a real-life application.

Later on, we focus on two autonomous driving related problems: traffic sign recognition and multi-object tracking and segmentation. For traffic sign recognition, in chapter 3 we take advantage from the hierarchical taxonomy of classes that traffic signs obey. The contributions of this chapter are summarized as:

- Informative novelty detection via semantic hierarchical taxonomies of classes.
- Metric learning based hierarchical novelty detection that enables assigning novel samples to their parent known classes by a distance based decision.
- Specific application to traffic sign recognition. We introduce the taxonomies and appropriate splits for two large scale traffic signs datasets, Mapillary Traffic Sign Dataset (MTSD) and Tsinghua-Tencent 100K (TT100K).
- New hierarchical novelty detection metric, *i.e.* the average error distance. It evaluates the errors produced under a hierarchical setting.

Finally, to mitigate the problem of the lack of labeled data in multi-object tracking and segmentation, in chapter 4 we have investigated a weakly supervised approach in terms of segmentation annotations. Its contributions are summarized as follow:

- A novel problem: weakly supervised multi-object tracking and segmentation, *i.e.* joint weakly supervised instance segmentation and multi-object tracking.
- New synergistic training strategy that takes advantage from multi-task learning, *i.e.* classification and tracking tasks guide the training of the unsupervised instance segmentation.

5.3 Future Work

The different problems covered throughout this thesis are still actively researched and have plenty of room for improvement.

We introduced a person re-identification pipeline in chapter 2, aiming at an efficient solution. There are still some aspects to improve from the perspective of a real-life application. This system might face the problem of domain adaptation. It refers to the difficulty of training networks with labeled datasets, that are then applied to new data recorded in different conditions, expecting them to still perform well. Also, in the direction of improving the efficiency of the person re-identification pipeline, the retrieval module can be further optimized. Since a brute-force search is needed to compare the person of interest against all the gallery images, this can lead to a bottleneck. The comparison time therefore increases with the size of the gallery, which is a real issue in a practical application. In this matter, a promising line of work is adding clustering and indexing techniques to reduce the computational cost at test time.

In chapter 3, we proposed a hierarchical novelty detection approach that learns class *prototypes* for every class in a semantic taxonomy of classes. As a future

line of research, our model could be applied to class incremental learning. By adding the detected novel classes at the proper taxonomy locations along with their corresponding class *prototypes* to our learned embedding, our model could be extended to recognize new classes in an incremental learning loop.

In a different line of work, we introduced the problem of weakly supervised multi-object tracking and segmentation in chapter 4. Although the problem of multi-object tracking and segmentation (MOTS) has been introduced recently, it has already attracted the attention of the research community. Weak forms of supervision for MOTS are still scarcely explored, however. Due to the difficulty of this joint task and its requirement of large amounts of annotated data to ensure a proper generalization, this is a promising direction to be explored in the forthcoming years. Weak supervision can be in the form of lacking either tracking or instance segmentation annotations. For unsupervised tracking, relying on optical flow could be an interesting direction to study. Regarding instance segmentation, the achieved performance on the predicted masks is still far from the fully supervised case. Strategies to improve the prediction at the borders of the objects would substantially improve the overall performance.

Finally, we believe some of the approaches presented in this dissertation might be complementary and combined towards solutions of more complex problem pipelines. For instance, combining person re-identification with novelty detection could lead to open-world person re-identification. A solution to such scenario would perform re-identification of known identities but should also recognize when it encounters a new individual then adding it to the dataset. Also, merging hierarchical novelty detection with MOTS could lead to an adaptive system that is suitable for autonomous driving open-world recognition. Such pipeline could be appropriate to automate annotation procedures or even design unsupervised approaches that include in their loop learning the newly detected classes.

List of Publications

1. **Idoia Ruiz**, Joan Serrat. Hierarchical Novelty Detection for Traffic Sign Recognition. Sensors, Volume 22, June 2022, 4389.

2. **Idoia Ruiz**, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, Joan Serrat. Weakly Supervised Multi-Object Tracking and Segmentation. In Proc. IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW). 2021.

3. Joan Serrat, **Idoia Ruiz**. Rank-based ordinal classification. In Proc. of International Conference on Pattern Recognition (ICPR). 2020.

4. Lorenzo Porzi, Markus Hofinger, **Idoia Ruiz**, Joan Serrat, Samuel Rota Bulò, Peter Kontschieder. Learning Multi-Object Tracking and Segmentation from Automatic Annotations. In Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). 2020.

5. **Idoia Ruiz**, Bogdan Raducanu, Rakesh Mehta, Jaume Amores, Optimizing speed/accuracy trade-off for person re-identification via knowledge distillation. Engineering Applications of Artificial Intelligence, Volume 87, January 2020, 103309.

6. Marc Masana, **Idoia Ruiz**, Joan Serrat, Joost van de Weijer, Antonio López. Metric Learning for Novelty and Anomaly Detection. In Proc. British Machine Vision Conference (BMVC). 2018.

7. Joan Serrat, Felipe Lumbreras, **Idoia Ruiz**. Learning to measure for preshipment garment sizing. Measurement, Volume 130, December 2018, Pages 327-339.

The content of this dissertation is based on the publications of which Idoia Ruiz is the first author.

A Appendix

A.1 Hierarchical Novelty Detection

A.1.1 Hyperparameters

All the hyperparameters were tuned by optimizing them to the validation set. Note that in hierarchical novelty detection there is no split of novel samples used as validation to search for hyperparameters. This, as discussed in [47], makes the problem more challenging, as we can only optimize our model to the validation set of the known classes, but the novel classes will always remain unknown.

State-of-the-art models. For the experiments on CUB and AWA, we use the parameters provided by the authors [47]. They train in a full-batch manner using an Adam optimizer with an initial learning rate of 10^{-2} and it decays at most two times when loss improvement is less than 2 compared to the last epoch. They apply L2 norm weight decay with parameter 10^{-2}.

For MTSD and TT100K we keep the same setting, except for the relabeling rate of the Relabel model, that was set to 15% and 30%, respectively.

HCL. For all the experiments on HCL, we use an Adam optimizer with a learning rate of 0.01 and the hyperparameters from Table A.1. In the experiments where we jointly train the backbone and HCL, the Adam optimizer uses a learning rate of 10^{-4} for the ResNet-101 backbone, for both datasets. Table A.1 contains: the regularization parameters for the HCL loss, $\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\}$, the batch size (BS), number of epochs (n_{epochs}) and relabeling rate r_{rate}. On the experiments of HCL being trained on precomputed features, we employ a full-batch training. The s parameter from the Normalized Softmax loss is always set to 40.

For the experiments in the ablation study, we keep the same hyperparameters as when we train the full version of HCL.

Table A.1: Hyperparameters to train HCL.

Experiment	$\{\lambda_{NS}, \lambda_{HC}, \lambda_{CT}, \lambda_{HT}\}$	BS	n_{epochs}	r_{rate}(%)
AWA2				
Imagenet feat.	$\{1, 1, 1, 0.01\}$	full-batch	1000	15
CUB				
Imagenet feat.	$\{1, 1, 1, 0.01\}$	full-batch	1000	15
TT100K				
Imagenet feat.	$\{1, 1, 1, 0.01\}$	full-batch	1000	30
Finetuned feat.	$\{1, 10, 1, 0.1\}$	full-batch	1000	30
Backbone + HCL	$\{1, 10, 1, 10\}$	280	300	30
MTSD				
Imagenet feat.	$\{1, 1, 1, 0.01\}$	full-batch	1000	15
Finetuned feat.	$\{1, 10, 1, 0.1\}$	full-batch	1000	15
Backbone + HCL	$\{1, 10, 1, 10\}$	280	300	15

A.1.2 Taxonomy figures

Taxonomies for TT100K, MTSD, AWA2 and CUB are depicted in Figures A.1, A.2, A.3 and A.4, respectively.

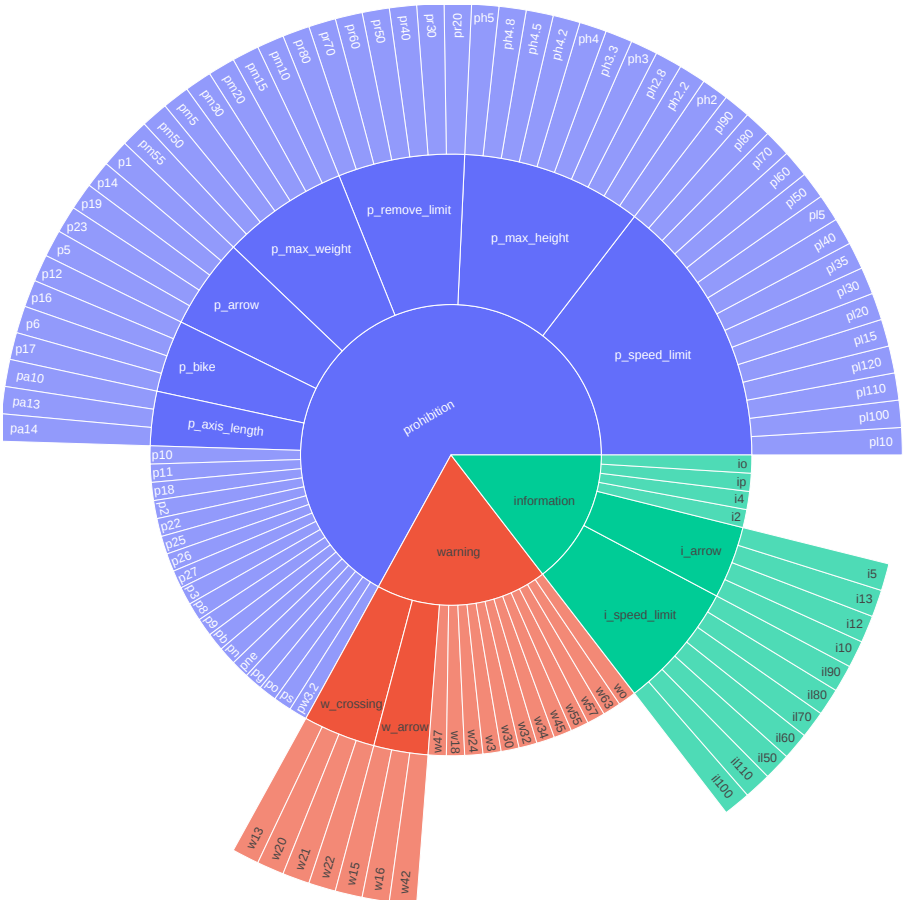


Figure A.1: TT100K class taxonomy. It contains both novel and known classes. See at full size at [79].

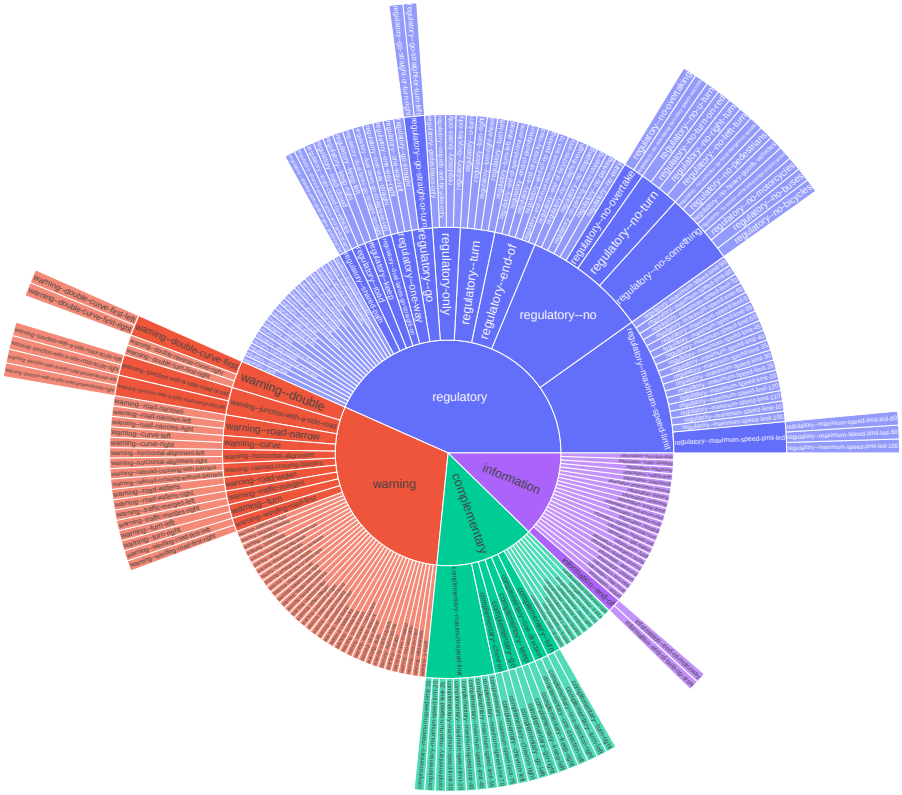


Figure A.2: MTSD class taxonomy. It contains both novel and known classes. See at full size at [79].

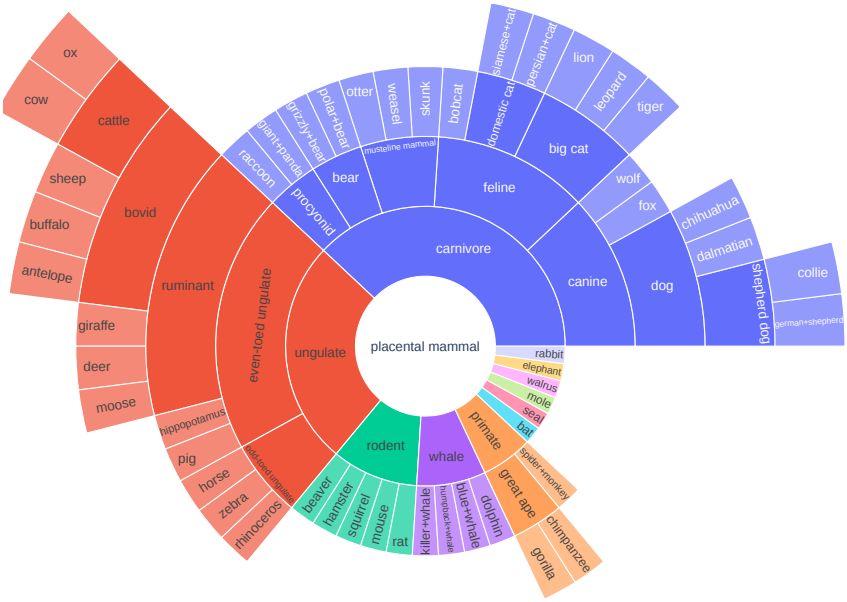


Figure A.3: AWA2 class taxonomy. It contains both novel and known classes. See at full size at [79].

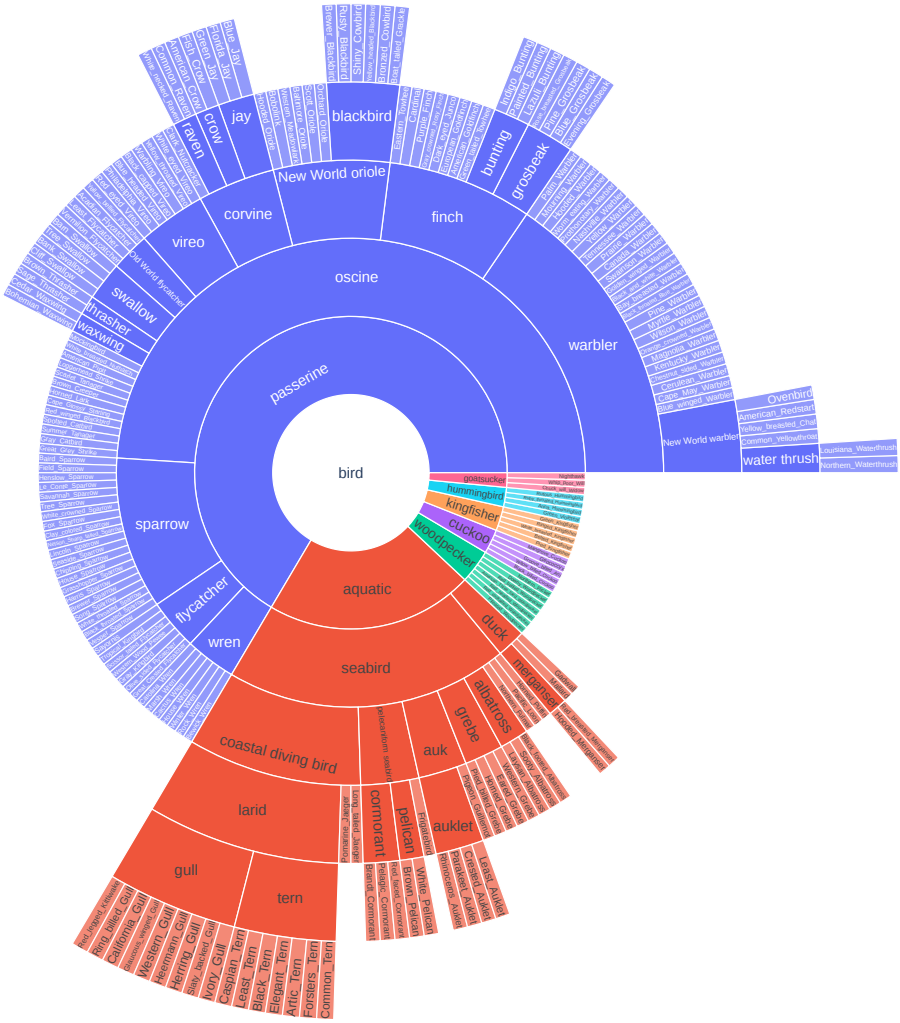


Figure A.4: CUB class taxonomy. It contains both novel and known classes. See at full size at [79].

Bibliography

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010.

[2] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3908–3916, Boston (MA), 2015.

[3] T.M. Feroz Ali and Subhasis Chaudhuri. Maximum margin metric learning over discriminative nullspace for person re-identification. In *European Conference on Computer Vision (ECCV)*, pages 123–141, Munich, Germany, 2018.

[4] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 441–450, 2017.

[5] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems (NIPS)*, pages 2654–2662, Montreal (Quebec), Canada, 2014.

[6] Slawomir Bak and Peter Carr. One-shot metric learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2990–2999, Honolulu (HI), 2017.

[7] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[8] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies

with deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12506–12515, 2020.

[10] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision (ECCV)*, pages 850–865. Springer, 2016.

[11] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.

[12] Terrance E Boult, Steve Cruz, Akshay Raj Dhamija, Manuel Gunther, James Henrydoss, and Walter J Scheirer. Learning and the unknown: Surveying steps toward open world recognition. In *AAAI Conference on Artificial Intelligence*, 2019.

[13] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 737–744, 1994.

[14] Clemens-Alexander Brust, Björn Barz, and Joachim Denzler. Making every label count: Handling semantic imprecision by integrating domain knowledge. In *International Conference on Pattern Recognition (ICPR)*, 2021.

[15] Clemens-Alexander Brust and Joachim Denzler. Integrating domain knowledge: Using hierarchies to improve deep classifiers. In *Asian Conference on Pattern Recognition (ACPR)*, 26-29 November 2019.

[16] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, Philadelphia (PA), 2006.

[17] Marco Buzzelli and Luca Segantin. Revisiting the compcars dataset for hierarchical car classification: New annotations, experiments, and results. *Sensors*, 21(2), 2021.

[18] Qiong Chen, Qingfa Liu, and Enlu Lin. A knowledge-guide hierarchical learning method for long-tailed image classification. *Neurocomputing*, 459:408–418, 2021.

[19] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–422, Honolulu (HI), 2017.

[20] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1335–1344, Las Vegas (NV), 2016.

[21] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7415–7424, 2018.

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

[23] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *European Conference on Computer Vision (ECCV)*, pages 71–84. Springer, 2010.

[24] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019.

[25] Weijian Deng, Liang Zheng, Guoliang Kang, Yi Yang, Qixiang Ye, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person reidentification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 994–1003, Salt Lake City (UT), 2018.

[26] Joachim Denzler, Erik Rodner, Paul Bodesheim, and Alexander Freytag. Beyond the closed-world assumption: The importance of novelty detection and open set recognition. In *German Conference on Pattern Recognition (GCPR) Unsolved Problems in Pattern Recognition Workshop (UPPR)*, 2013.

[27] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, and Yubin Kuang. Traffic sign detection and classification around the world. In *European Conference on Computer Vision (ECCV)*, 2020.

[28] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[29] Vivien Sainte Fare Garnot and Loic Landrieu. Leveraging class hierarchies with metric-guided prototype learning. In *British Machine Vision Conference (BMVC)*, 2021.

[30] Yixiao Ge, Zhuowan Li, Haiyu Zhao, Guojun Yin, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Fd-gan: Pose-guided feature distilling gan for robust person re-identification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–12, Montreal, Canada, 2018.

[31] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 43(10):3614–3631, 2020.

[32] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE, 2006.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas (NV), 2016.

[34] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[35] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[36] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[37] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[38] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[39] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Scalability in Autonomous Driving*, 2020.

[40] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 41(3):523–536, 2018.

[41] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[42] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2295, Providence (RI, 2012.

[43] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 695–711. Springer, 2016.

[44] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*, pages 109–117, 2011.

[45] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning (ICML)*, pages 513–521, 2013.

[46] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013.

[47] Kibok Lee, Kimin Lee, Kyle Min, Yuting Zhang, Jinwoo Shin, and Honglak Lee. Hierarchical novelty detection for visual object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[48] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations (ICLR)*, 2018.

[49] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[50] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4282–4291, 2019.

[51] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8971–8980, 2018.

[52] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 384–393, Honolulu (HI), 2017.

[53] Qizhu Li, Anurag Arnab, and Philip HS Torr. Weakly-and semi-supervised panoptic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 102–118, 2018.

[54] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Columbus (OH), 2014.

[55] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2197–2206, Boston (MA), 2015.

[56] Shengcai Liao, Guoying Zhao, Vili Kellokumpu, Matti Pietikäinen, and Stan Z Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1301–1306. IEEE, 2010.

[57] Chung-Ching Lin, Ying Hung, Rogerio Feris, and Linglin He. Video instance segmentation tracking with a modified vae architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[58] Ji Lin, Liangliang Ren, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Consistent-aware deep learning for person re-identification in a camera network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3396–3405, 2017.

[59] Shan Lin, Haoliang Li, Chang-Tsun Li, and Alex Chichung Kot. Multi-task mid-level feature alignment network for unsupervised cross-dataset person

re-identification. In *British Machine Vision Conference (BMVC)*, pages 1–13, Newcastle, UK, 2018.

[60] Jiawei Liu, Zheng-Jun Zha, Di Chen, Richang Hong, and Meng Wang. Adaptive transfer network for cross-domain person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7202–7211, Long Beach (CA), 2019.

[61] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 212–220, 2017.

[62] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, volume 2, page 7, 2016.

[63] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2):1803–1810, 2020.

[64] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. Unovost: Unsupervised offline video object segmentation and tracking. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2000–2009, 2020.

[65] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, and Xiaoou Tang. Face model compression by distilling knowledge from neurons. In *AAAI Conference on Artificial Intelligence*, pages 3560–3566, Phoenix (AZ), 2016.

[66] Marc Masana, Idoia Ruiz, Joan Serrat, Joost van de Weijer, and Antonio M Lopez. Metric learning for novelty and anomaly detection. In *British Machine Vision Conference (BMVC)*, 2018.

[67] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptor for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1363–1372, Las Vegas (NV), 2016.

[68] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5382–5390, 2017.

[69] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012, 2016.

[70] Rameswar Panda, Amran Bhuiyan, Vittorio Murino, and Amit K. Roy-Chowdhury. Unsupervised adaptive re-identification in open world dynamic camera networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1377–1386, Honolulu (HI), 2017.

[71] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):1–38, 2021.

[72] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391, San Francisco (CA), 2010.

[73] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. *arXiv preprint arXiv:1905.01220*, 2019.

[74] Lorenzo Porzi, Markus Hofinger, Idoia Ruiz, Joan Serrat, Samuel Rota Bulo, and Peter Kontschieder. Learning multi-object tracking and segmentation from automatic annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6846–6855, 2020.

[75] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *European Conference on Computer Vision (ECCV)*, pages 650–667, Munich, Germany, 2018.

[76] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. 2016.

[77] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision (ECCV) Workshops*, pages 1–18, Amsterdam, The Netherlands, 2016.

[78] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning and Representation (ICLR)*, pages 1–10, San Diego (CA), 2015.

[79] Idoia Ruiz. Github repository. Available online: https://github.com/idoiaruiz/HierarchicalCosineLoss.git. (accessed on 21 April 2022).

[80] Idoia Ruiz, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Joan Serrat. Weakly supervised multi-object tracking and segmentation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 125–133, January 2021.

[81] Idoia Ruiz, Bogdan Raducanu, Rakesh Mehta, and Jaume Amores. Optimizing speed/accuracy trade-off for person re-identification via knowledge distillation. *Engineering Applications of Artificial Intelligence*, 87:103309, 2020.

[82] Idoia Ruiz and Joan Serrat. Hierarchical novelty detection for traffic sign recognition. *Sensors*, 22(12), 2022.

[83] Fatemehsadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, Stephen Gould, and Jose M Alvarez. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 413–432. Springer, 2016.

[84] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, Boston (MA), 2015.

[85] Arne Schumann and Rainer Stiefelhagen. Person re-identification by deep learning attribute-complementary information. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1435–1443, Honolulu (HI), 2017.

[86] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[87] Sanghyun Seo and Juntae Kim. Hierarchical semantic loss and confidence estimator for visual-semantic embedding-based zero-shot learning. *Applied Sciences*, 9(15), 2019.

[88] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in Neural Information Processing Systems (NIPS)*, 29, 2016.

[89] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5620–5629, 2017.

[90] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3136–3145, 2019.

[91] Jifei Song, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. Generalizable person re-identification by domain-invariant mapping network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 719–728, Long Beach (CA), 2019.

[92] Juan Luis Suárez, Salvador García, and Francisco Herrera. A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. *Neurocomputing*, 425:300–322, 2021.

[93] Mingjie Sun, Jimin Xiao, Eng Gee Lim, Bingfeng Zhang, and Yao Zhao. Fast template matching and update for video object tracking and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[94] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3800–3808, Honolulu (HI), 2017.

[95] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *European Conference on Computer Vision (ECCV)*, September 2018.

[96] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *European Conference on Computer Vision (ECCV)*, pages 135–153, Amsterdam, The Netherlands, 2016.

[97] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[98] Safat B. Wali, Majid A. Abdullah, Mahammad A. Hannan, Aini Hussain, Salina A. Samad, Pin J. Ker, and Muhamad Bin Mansor. Vision-based traffic sign detection and recognition systems: Current trends and challenges. *Sensors*, 19(9), 2019.

[99] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *ACM International Conference on Multimedia*, pages 1041–1049, 2017.

[100] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5265–5274, 2018.

[101] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2275–2284, Salt Lake City (UT), 2018.

[102] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 79–88, Salt Lake City (UT), 2018.

[103] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[104] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[105] Ancong Wu, Wei-Shi Zheng, Xiaowei Guo, and Jian-Huang Lai. Distilled person re-identification: Towards a more scalable system. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1187–1196, Long Beach (CA), 2019.

[106] Tz-Ying Wu, Pedro Morgado, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos. Solving long-tailed recognition with deep realistic taxonomic classifier. In *European Conference on Computer Vision (ECCV)*, pages 171–189. Springer, 2020.

[107] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 41(9):2251–2265, 2018.

[108] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[109] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1249–1258, Las Vegas (NV), 2016.

[110] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang. Segment as points for efficient online multi-object tracking and segmentation. In *European Conference on Computer Vision (ECCV)*, 2020.

[111] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Xiangbo Su, Yuchen Yuan, Hongwu Zhang, Shilei Wen, Errui Ding, and Liusheng Huang. Pointtrack++ for effective online multi-object tracking and segmentation. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Benchmarking Multi-Target Tracking: Multi-Object Tracking and Segmentation*, 2020.

[112] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5188–5197, 2019.

[113] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *International Conference on Pattern Recognition (ICPR)*, pages 34–39, Stockholm, Sweden, 2014.

[114] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, pages 1–11, Toulon, France, 2017.

[115] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10823–10832, 2019.

[116] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4320–4328, Salt Lake City (UT), 2018.

[117] Yingying Zhang, Qiaoyong Zhong, Linda Ma, Di Xie, and Shiliang Pu. Learning incremental triplet margin for person re-identification. In *AAAI Conference on Artificial Intelligence*, pages 1–8, Honolului (HI), 2019.

[118] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[119] Kai Zhao, Jingyi Xu, and Ming-Ming Cheng. Regularface: Deep face recognition via exclusive regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[120] Feng Zheng, Cheng Deng, Xing Sun, Xinyang Jiang, Xiaowei Guo, Zongqiao Yu, Feiyue Huang, and Rongrong Ji. Pyramidal person re-identification via multi-loss dynamic training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8514–8522, Long Beach (CA), 2019.

[121] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, pages 1116–1124, Boston (MA), 2015.

[122] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

[123] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, and Qi Tian. Person re-identification in the wild. In *IEEE International Conference on Computer Vision*, pages 1367–1376, 2017.

[124] Meng Zheng, Srikrishna Karanam, Ziyan Wu, and Richard J. Radke. Re-identification with consistent attentive siamese networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5735–5744, Long Beach (CA), 2019.

[125] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[126] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Associating groups of people. In *British Machine Vision Conference (BMVC)*, pages 1–11, Cardiff, UK, 2009.

[127] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2138–2147, Long Beach (CA), 2019.

[128] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *IEEE International Conference on Computer Vision*, pages 3754–3762, Venice, Italy, 2017.

[129] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1318–1327, Honolulu (HI), 2017.

[130] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. Generalizing a person retrieval model hetero-and homogeneously. In *European Conference on Computer Vision (ECCV)*, pages 176–192, Munich, Germany, 2018.

[131] Da-Wei Zhou, Yang Yang, and De-Chuan Zhan. Learning to classify with incremental new class. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[132] Sanping Zhou, Jinjun Wang, Jiayun Wang, Yihong Gong, and Nanning Zheng. Point to set similarity based deep feature learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5028–5037, Honolulu (HI), 2017.

[133] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2110–2118, 2016.