Deep Learning for Chairlift Scene Analysis : Boosting Generalization in Multi-Domain Context

Hiba Alqasir

Apprentissage profond pour l'analyse de scènes de remontées mécaniques: amélioration de la généralisation dans un contexte multi-domaines

Deep Learning for Chairlift Scene Analysis: Boosting Generalization in Multi-Domain Context

N°d'ordre NNT: 2020LYSES045
THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
Université Jean Monnet de Saint-Étienne
Laboratoire Hubert Curien

École Doctorale ED SIS n°488
École Doctorale Science, Ingénierie et Santé

Spécialité de doctorat: Informatique
Discipline: Vision par ordinateur

Thèse préparée par:

Hiba ALQASIR

Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d'Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, Saint-Etienne, France.

Soutenue à huis clos le **17 Decembre 2020**, devant le jury composé de:

Nicolas Thome	Professeur, CNAM, Laboratoire Cédric	Rapporteur
Joost van de Weijer	Senior Scientist, Universitat Autònoma de Barcelona, CVC	Rapporteur
Michèle Rombaut	Professeure, Université Grenoble Alpes, GIPSA-Lab	Examinatrice
Elisa Fromont	Professeure, Université de Rennes 1, IRISA/INRIA	Examinatrice
Raluca Debusschere	Docteur, Bluecime	Examinatrice
Christophe Ducottet	Professeur, Université Jean-Monnet, Hubert Curien	Directeur
Damien Muselet	Maître de conférence, Université Jean-Monnet, Hubert Curien	Encadrant

We're all human, aren't we? Every human life is worth the same, and worth saving."

—J.K. Rowling, *Harry Potter and the Deathly Hallows*

Abstract

This thesis presents our work on chairlift safety using deep learning techniques as part of Mivao project, which aims to develop a computer vision system that acquires images of the chairlift boarding station, analyzes the crucial elements, and detects dangerous situations. In this scenario, different chairlifts spread over different ski resorts, with a high diversity of acquisition conditions and geometries; thus, each chairlift is considered a domain. When the system is installed for a new chairlift, the objective is to perform an accurate and reliable scene analysis, given the lack of labeled data on this new domain (chairlift).

In this context, we mainly concentrate on the chairlift safety bar and propose to classify each image into two categories, depending on whether the safety bar is closed (safe) or open (unsafe). Thus, it is an image classification problem with three specific features: (i) the image category depends on a small detail (the safety bar) in a cluttered background, (ii) manual annotations are not easy to obtain, (iii) a classifier trained on some chairlifts should provide good results on a new one (generalization). To guide the classifier towards the important regions of the images, we have proposed two solutions: object detection and Siamese networks. Furthermore, we analyzed the generalization property of these two approaches.

Object detection is a solution that allows guiding the model towards the crucial regions of the image. At test time, we can use it as a classification model by ignoring the location of the detected instances. In this thesis, we show that such an approach helps increase the classification accuracy for our specific problem compared to a classical classification network. The main weakness is the need for bounding box annotations to train the model. Thus, we have proposed to automatically create bounding box annotations by exploiting the results of an existing algorithm based on hand-crafted features. Then, since the safety bar geometry varies across chairlifts, we have proposed an original unsupervised domain adaptation step designed for object detection.

The second solution we have proposed to guide the classification model is to insert shape priors in the model using a Siamese network. The idea consists in providing pairs of images to the network, one colored image of the chairlift and one binary mask representing the safety bar. This solution requires only two masks for each chairlift, which we could obtain easily. If the safety bar in the colored image and binary mask are both open or both closed, we enforce the network to extract similar features from the image and the mask, while when the safety bar is open (resp. closed) in the colored image and closed (resp. open) in the binary mask, the features have to be different. This approach forces the network to concentrate on the safety bar in the image while ignoring the background. Furthermore, we show that specific virtual masks boost the generalization property of the network for new unseen chairlifts without requiring any images from these chairlifts.

Our solutions are motivated by the need to minimize human annotation efforts while improving the accuracy of the chairlift safety problem. However, these contributions are not necessarily limited to this specific application context, and they may be applied to other problems in a multi-domain context.

Keywords

Deep learning . Chairlift safety . Generalization . Multi-domain . Object deletion . Image classification . Domain adaptation . Siamese Networks .

Acknowledgments

It has been a great adventure to get my thesis completed over the past three years. Such a memorable, insightful and exciting trip. Well, except that the world went into a chaotic state where the number of COVID-19 cases has been increasing, businesses were forced to close and people were scrambling to stock up on as many rolls of toilet paper as possible, I still had a great time pursuing my degree. For that, I want to thank those on the front lines for keeping our lives going despite the pandemic.

Then and above all, I would like to thank my supervisors for their precious advice and patience during my doctoral studies. Christophe Ducottet, whose invaluable expertise was indispensable in shaping the research questions and methodology. Damien Muselet, whose guidance and assistance have encouraged me at every stage of this journey. Thank you both for providing me with the tools I needed to choose the right direction to complete my thesis. Our countless emails and meetings have helped me sharpen my thinking and take my work to the next level.

I want to express my sincere gratitude to my thesis reporters Nicolas Thome and Joost van de Weijer, for their enriching review of the manuscript. And to each member of my thesis committee who despite the current public health situation, managed to attend in person or remotely, to ensure that the defense could be held. Thank you for your perceptive comments and challenging questions that inspired me to broaden my research from different angles.

To all those I have enjoyed working with, my lab mates, colleagues, and the research team, I express my thanks for your technical, scientific, and moral support. I also would like to acknowledge Bluecime team for their outstanding collaboration and valuable feedback and suggestions. I would particularly like to single out Raluca Debusschere, for her generous support, and for all of the discussions we had to further my research.

While supervision, guidance, and discussion have all been important in making this thesis a reality, some people have also made a direct contribution to its realization, and indeed to the realization of everything I have accomplished in my life: my beloved family. I can not thank my Mom and Dad enough or be grateful to them sufficiently for their love, encouragement, and unwavering confidence in me. Also, I would not have been able to complete this thesis without the backing of my sister and brother. They provided me with stimulating discussions and happy distractions to rest my mind outside of my studies. I must also thank my family-in-law for their emotional support and heartwarming care.

I can not forget my friends who have cheered me on through hard times and celebrated every little accomplishment with me. I have been fortunate to have a wonderful bunch of friends here in France who have been my second family. To my many other friends worldwide, thank you for your thoughts, wishes, phone calls, messages, chats, visits, and for being there whenever I needed you.

Finally, my research and lifelong partner, Dr. (to be) Alaa Daoud: you are my rock through this journey and all the others. You have always been my cheerleader and my lifeline when things got a little discouraging. Without your great understanding and enthusiasm, my life would not be the same. For this, and for all the beautiful days we have had and will have together, I am deeply grateful.

Contents

List of Figures

List of Tables

Introduction

Ski resorts are a favorite winter attraction in almost every country that has mountainous regions and winter seasons. France is Europe's most popular ski destination; it has the largest fleet of chairlifts in the world (counting more than 3000). Chairlifts are not used only for ski in winter, but also for the rest of the year to transport passengers from valleys to mountain peaks, to practice other mountain sports such as cycling or hiking. Among millions of chairlifts trips per year, only a few accidents happen (28 accidents out of the 403 million chairlifts trips in France for the season 2018/2019, according to the report on accidents on ski lifts in France and concerning users transported during the 2018/2019 season in [4]), but they are potentially severe and sometimes fatal. Many factors influence the accident risk, including weather conditions, traffic on the slopes, passenger experience and skills, *etc.*

Nevertheless, most accidents are due to human error and risky behavior. Chairlifts have a safety bar (or restraining bar) to safely and adequately maintain passengers in their seats. When passengers board the chairlift, they must grab the safety bar and pull it to its lower limit, and before getting off, they must pull it up, making it the passengers' job to open and close the safety bar (see Figure 1). However, resorts have a control system monitoring the chairlifts at boarding to ensure that passengers have correctly closed the safety bar and in case they did not, it warns them. This monitoring system typically consists of human operators who monitor and detect any dangerous situation when boarding and upon arrival.

Figure 1 – Chairlifts leaving the boarding station in three situations[1], from left to right: (i) passengers have fully closed the safety bar (*safe*), (ii) passengers are closing the bar (*unsafe*), (iii) passengers did not close the bar (*unsafe*).

To increase the safety on chairlifts, the french start-up Bluecime developed Sivao [2], a computer vision system to help the ski lift operators with insulating problems by detecting anomalies when boarding the chairlift, to warn the users in dangerous situations and prevent accidents eventually. The system consists of a camera, a computer, and an alarm. The camera records the boarding scene. Then certain video frames are processed in real-time using non-learning-based image processing techniques to detect: (i) chairlift occupancy; (ii) safety bar position. Whenever a hazardous situation is detected *i.e.*, the chairlift is occupied, and the safety bar is not fully closed, the alarm is triggered.

[1] The people's faces in the figures of this thesis have been intentionally blurred, for reasons of privacy.
[2] Sivao: Système Intelligent de Vision Artificielle par Ordinateur.

Although the decisions made by SIVAO are very accurate, the installation for a new chairlift requires costly and time-consuming manual configurations. As an increasing number of chairlifts are equipped with the system, considerable time and effort are required to configure the installation. Hence, the idea of MIVAO [3] project is to use deep learning to improve the performance of SIVAO and decrease the costly manual adjustments for different environments and circumstances while also benefiting from the increasing amount of data at hand.

The latest advancements in deep learning-based models brought original solutions that attract the attention of the industry. Approaches based on deep Convolutional Neural Networks (CNNs) showed substantial improvements in the generalization of a wide range of problems, varying from simple image recognition tasks, such as handwritten digit-recognition (early beginning of CNNs), to more complex tasks such as predicting the impact of changing non-coding DNA sequences in gene expression. The key advantage of deep learning methods is that they learn high-level features from the data automatically and progressively, using a general-purpose learning procedure. Deep learning methods are powered by a huge amount of data, while they only require nominal manual pre-processing and negligible feature engineering, which negates the need for expertise in the field, contrary to traditional machine learning techniques. Figure 2 illustrates the main difference between deep learning and traditional machine learning methods, which is the challenging feature extraction step.

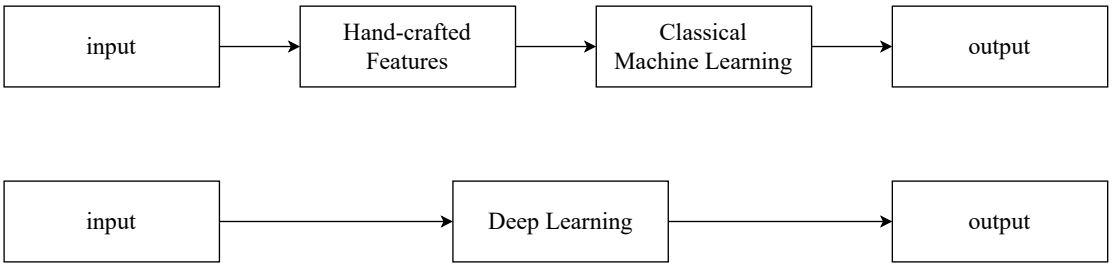


Figure 2 – Deep learning vs. traditional machine learning methods. Deep learning methods learn high-level features from the data automatically.

This thesis is a part of MIVAO project to secure the boarding on chairlifts. We address this problem by proposing deep learning techniques to enhance SIVAO, and exploit the available data.

Research problems

MIVAO aims to create an intelligent system to perform an extensive analysis of the boarding scene and identify rare but potentially hazardous situations in chairlifts. Comprising SIVAO functionalities (*i.e.* real-time detection of the chairlift occupancy and the position of the safety bar), MIVAO is expected to be more reliable and less sensitive to changing conditions, taking into account the reduction of the configuration workload when installing the system in new locations.

Almost all chairlifts have human operators monitoring the boarding station; these operators' job is to monitor and react to anomalous situations. However, continuous manual monitoring is laborious for humans, so the automatic analysis of surveillance videos is a solution to this type of tedious tasks. The analysis of the surveillance videos consists mainly of two levels: (i) low-level where essential elements are detected, and (ii) high-level where the results of the low-level are used for decision making. The low-level analysis involves different image and video analysis algorithms from research areas like classification, object detection, object tracking, action recognition, *etc*. In the chairlift safety problem, the essential elements are people (it is crucial to differentiate between adults and children) and the safety bar (it is imperative to determine its position: completely open, completely closed, or between). Therefore, in this thesis, we focus on adult/child and open/closed safety bar detection and classification.

[3] MIVAO: Montagne Innovante, Vision et Apprentissage par Ordinateur

To meet MIVAO's challenges, we attempt to address the issue of exploiting the available data to train robust deep learning models with the minimal amount of annotations possible, taking into account the geometrical constraints to improve detection. Accordingly, we opt to investigate unsupervised domain adaptive object detection and image classification. First of all, it is pertinent to clarify the occasional confusion between image classification and object detection tasks. *Image classification* refers to the task of naming the dominant object in an image. On the other hand, *Object detection* is to name and localize all the objects in the image. Figure 3 shows an example of the expected output in each task.

Image class: open safety bar.

(a) Image classification: image classified based on only one object.

Objects: open safety bar, and two adults.

(b) Object Detection: three objects named and localized with bounding boxes.

Figure 3 – Image classification vs. Object detection in chairlift safety problem.

Supervised learning entails training the system by rewarding it if it gives the expected output for a given input and penalizing it if it does not. Image classifiers are intended to output the class of the image; therefore, they are trained using image-level annotations. On the other side, object detectors require bounding box annotations around every object in the training images. These annotations are essential for training in a supervised manner, but at the same time, they are expensive and time-consuming as they are usually generated manually.

Predicting the output for new inputs on which the system has not been trained is called *generalization*. If the training data is representative of the underlying distribution, then the system will generalize well. However, if the inputs at the time of test are significantly inconsistent with the training data, the model may not generalize well. For this reason, generalizing towards wider distributions seems rather too far-reaching, whereas targeting a particular distribution is more achievable. This problem area is called *domain adaptation*. The training data is used as a source of additional information for the target domain. If no labels are available in the target domain, it is known as *unsupervised domain adaptation*. The challenge is to overcome the difference between domains, to the extent that a system trained in the source domain will generalize well to the target domain.

The core of this thesis focuses on the challenge of generalizing to new domains with a minimum amount of annotations by exploiting geometrical constraints. In this problem, each chairlift is considered a new domain, due to the significant variations between chairlifts in terms of shape, size, number of seats, orientation, *etc.* (see Figure 4 for examples).

Figure 4 – Examples of variations in shape and appearance of four different chairlifts *i.e.* (from left to right respectively) C_{12}, C_2, C_{18}, and C_1. We don't give chairlifts real names for confidentiality reasons.

Motivation and Contributions

In the context of MIVAO project, a large collection of images was available without annotations. Our objective was to seek an efficient way to use this data with the least amount of annotations. In this thesis, we focus our attention on unsupervised domain adaptation in object detection and image classification. We propose solutions to improve the learning process and therefore the accuracy, to generalize and achieve competitive performance where few or no labeled data is available.

An initial contribution of this thesis was to apply a state of the art CNN-based method for object detection on the chairlifts dataset to evaluate its ability to detect the safety bar and the people, to spot risky situations eventually. As expected, this approach provides better results than a classical classifier trained without using bounding box annotations. However, training these models requires a large number of annotations, yet one of the most important objectives of the project is to find a self-configuring solution when applied to new unlabelled data. Therefore, **domain adaptation in object detection** is the first area studied in the thesis. We focus on the question: what features should be adapted in an object detector to generalize from a source to a target domain? Few works explicitly address the problem of unsupervised domain adaptation for object detection. Existing approaches added adversarial training components in the classical Faster R-CNN detector, at both global and instance levels without adapting the Region Proposal Network (RPN), leading to a residual domain shift. We proposed to **adapt the RPN** to ensure that the features extracted from the target images overlap with the source object features.

However, even only annotating the source domain is very expensive when it comes to instance-level bounding boxes. Besides, domain adaptation is not always satisfactory at the scale of dozens of different target domains; the most critical success factor of the domain adaptation is the level of similarity between the source and target domains. For the second contribution addressing the expense of bounding boxes annotations, we opted for the **use of binary masks to guide a classifier**. We concentrate on the issue: how to exploit geometrical constraints as priors to design an image classifier and train it on only a few labeled images? The main idea is to inform the classifier of the element on which it should focus in the image to take its decision, by using a binary mask, eliminating the need for much more expensive bounding box annotations. Using a Siamese architecture, we feed a CNN-based image classifier with the image and the binary mask, and we force the classifier to extract only the requisite features according to the binary mask.

When the attempt was successful, the question was: is this classifier able to generalize? Moreover, how to improve its generalizability? We propose a fine-tuning step that uses only two masks from the target domain and provides outstanding results without requiring any specific domain adaptation component. Furthermore, we show that specific virtual masks boost the generalization property of the network for new unseen chairlifts without requiring any images nor masks from these target chairlifts. Besides, this last approach also outperforms object detection without any need for instance-level bounding box annotations.

As mentioned above, the principal use case considered in this thesis is the chairlift safety problem. Although the motivation of our work is to address the problems of MIVAO, the contributions made are not limited to this context, but could be applied to any application requiring scene analysis, and could be useful in several scenarios such as autonomous driving, as we will show in the experiments.

Structure of the dissertation

This thesis aims to propose methods based on deep learning to identify dangerous situations in chairlifts. These methods are intended to improve the performance of the existing SIVAO system by enhancing the accuracy and reducing the time needed to configure the system for a new chairlift. The remaining chapters of this manuscript are organized as follows:

- Chapter 1 presents MIVAO project, its objectives, challenges, and perspectives, and provides a detailed description of its dataset. Moreover, it presents our MIVAO-related objectives in this thesis, being a part of the project.
- Chapter 2 provides an overview of the available literature on the main aspects related to our objectives. In this chapter, we review the solutions that already exist in the field of chairlift safety. Besides, we study image classification methods based on deep learning. Then, we present object detection and argue its main advantages for the chairlift safety problem. The next aspect to consider is domain adaptation. Finally, we review what has already been proposed to exploit a particular prior knowledge in deep learning models.
- Chapter 3 includes a preliminary contribution: applying object detection to the chairlift safety problem. This chapter evaluates the performance of the state of the art Faster R-CNN approach on the task and outlines its main limitations.
- Chapter 4 presents a new viewpoint about the domain shift problem in object detection. We propose to adapt the Region Proposal Network (RPN) and integrate this new adaptation module in Faster R-CNN. In addition to the chairlift safety problem, we conduct experiments in the context of autonomous driving to compare our proposed approach with another unsupervised domain adaptation method applied to Faster R-CNN.
- Chapter 5 presents a solution for CNN-based image classification task where the class of each image depends on a small detail in it. We propose to use Siamese networks to solve image classification problem with geometrical constraints as priors.
- Chapter 6 is dedicated to present a solution to improve the learning process of a classification network when less labeled images are required. We study the generalization power of such an approach, and we propose a solution to improve it.
- Finally, in conclusion, we summarize the work presented in the previous chapters. After that, we discuss some of the possible future work perspectives and research directions.

Chapter 1

Context: Chairlift Safety Problem

The safety of chairlifts is a major concern for ski resort operators. To prevent possible accidents, it is necessary to detect dangerous situations on chairlifts as early (after boarding) as possible. Mivao project was launched based on the needs of ski resort operators to secure chairlifts. Its main goal is to develop a machine learning and computer vision system to do real-time analysis of the boarding scene. This chapter is dedicated to present the project, its objectives, challenges, and perspectives, and to describe its dataset in detail.

1.1 MIVAO project

First, it is imperative to introduce Sivao: the system Bluecime has been developing since 2015, which aims to assist ski lift operators with insulating rare but potentially dangerous situations to increase the safety of chairlifts. The system acquires images from the boarding station of chairlifts, analyzes the important elements (people, chairlift carrier, safety bar, *etc.*) in real-time, and detects hazardous conditions during chairlift boarding. In ski resorts where this system is used, a camera is fixed on one of the first pillars of the chairlift to track it in a detection zone, as illustrated in Figure 1.1.

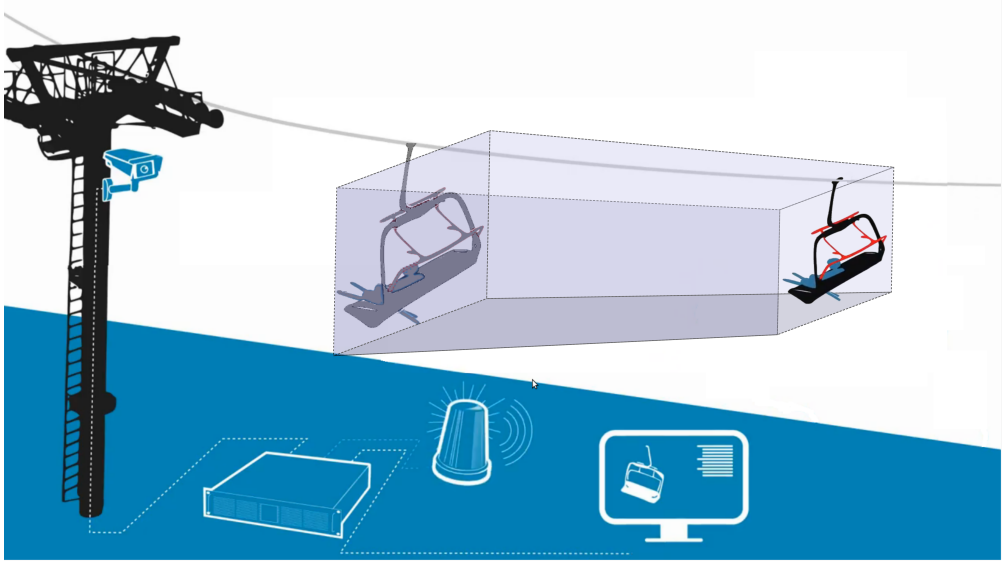


Figure 1.1 – Sivao system and detection zone.

The system was using non-learning-based image processing techniques [1] to detect if the chairlift is empty or not and if its safety bar is open or closed. At the end of the detection zone,

[1] In the context of Mivao, Bluecime has already integrated deep-learning processes to optimize and simplify some configurations (*e.g.*, people counting), and the work is still in progress.

if passengers did not completely close the safety bar (the chairlift is not empty and the safety bar is open), the system triggers an alarm. The accuracy of decisions made by the system is very high; however, its weak point is that installing the system for a new chairlift requires costly and time-consuming manual configurations.

The system is installed today in 22 ski resorts with 54 different chairlifts [2], and the number is growing, resulting in a huge amount of data on one hand, and a great effort to monitor, maintain the system or install it in a new station on the other hand. Hence the idea of MIVAO project is to use machine learning methods and, in particular, those based on deep learning, to improve the performance of SIVAO and reduce the expensive manual adjustments for varying conditions. MIVAO project is led by BLUECIME in partnership with academic partners *i.e.* HUBERT CURIEN LABORATORY and GIPSA-LAB, as well as industrial partners *i.e.* SOFIVAL and DSV which both manage ski resorts in the french Alpes. This thesis is part of the project along with two others and two post-docs, each approaches the problem of chairlift safety from a different, but complementary, aspect.

MIVAO aims to develop a system to provide an in-depth analysis of the boarding scene and detect not only the users who have not appropriately closed the safety bar, but also more sophisticated situations like a user slipped from his carrier or the presence of unaccompanied children, *etc.* This information has to be reported to the operating staff and the users, using light and sound alarms. In the long term, if the success rate is sufficient, it is possible to go towards automating the chairlift to stop or slow down the mechanism automatically in case of danger. By and large, the project aims to develop a system that will operate whatever the weather conditions and self-configure according to the context during installation. The objectives to be achieved for the reliability of the detection of dangerous situations are the following:

- robustness against the variability in shape and appearance of the chairlifts in the ski resorts;
- robustness against weather conditions variations such as rain, snow, fog, *etc.* (see Figure 1.2 for some examples);
- robustness against lighting conditions variations during the day (see Figure 1.3 for some examples);
- robustness against the variability of users' outfits and accessories like hats, helmets, ski glasses, *etc.*;
- automate the system configuration phase to minimize installation costs;
- master real-time constraints, despite the use of robust algorithms.

Challenges and motivations. Although there is a substantial base of knowledge in the image field in the scientific community and the industrial world, the application of image analysis techniques for the specific problem of "monitoring chairlifts to improve safety" presents major technological challenges. Below we show the most important of them:

- Studying the behavior of passengers to detect risky situations raises critical technical difficulties. Conventionally, to interpret a person's behavior, it is necessary to perform a preliminary phase of a silhouette or face detection, which is very difficult in a project like MIVAO, where passengers are hidden by ski equipment in winter or mountain bike equipment in summer, and often have their faces entirely or partially concealed. Likewise, accurately detecting the position of a safety element (*e.g.* the safety bar on a chairlift carrier, that should be closed when the carrier is occupied) can be very difficult in some cases (the bar can be hidden by the arms of the passengers or by the sports equipment they carry on the vehicle).
- The chairlifts move at 5m/s, and spaced at distances of 5 to 40 meters. This implies the need for fast and efficient treatment in real-time.
- Installation and configuration of the system for a new chairlift should be quick and easy; that is to say, the intervention of highly qualified staff should be limited.

[2] In MIVAO, only data from partner stations are accessible, *i.e.* 25 chairlifts in 2020.

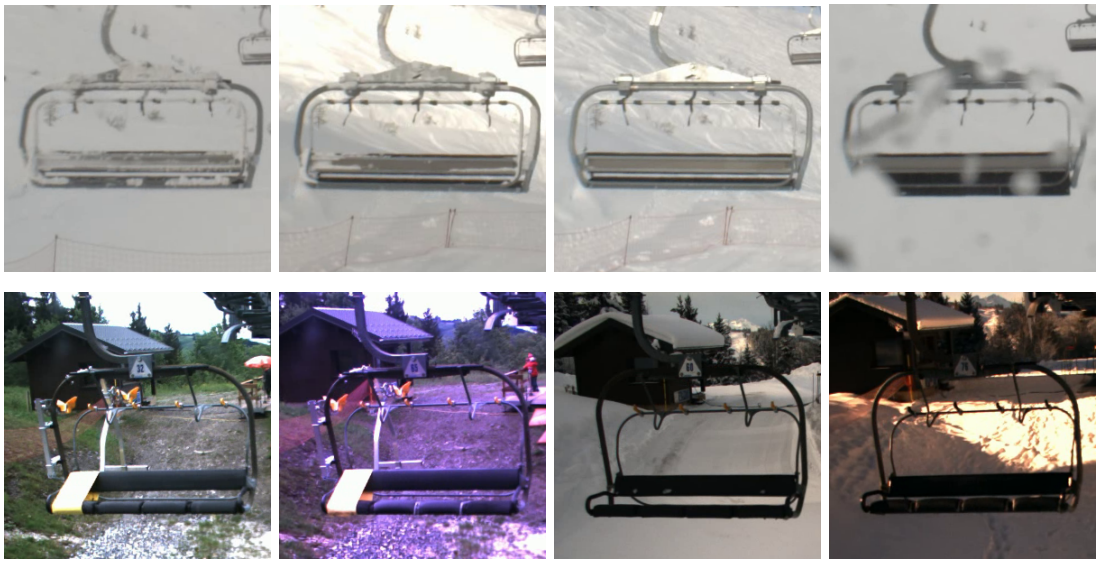


Figure 1.2 – Examples of variations in weather conditions in two different chairlifts, namely C_{12} (first row) and C_{20} (second row).

Traditional computer vision techniques have shown their limitations during intensive test campaigns carried out by Bluecime during winter 2015-2016 to achieve Mivao objectives. Given the expertise of each of the project partners and the amount of available data (which has increased over the project), machine learning methods and, in particular, those based on deep learning, would make it possible to more precisely determine the risk linked to a given situation on a chairlift and would be able to respond satisfactorily to the challenges of the project.

The use of machine learning methods involves first creating a model that describes the problem to be solved. Learning such a model will be done in offline mode, without the computation time posing a concern. Therefore, the real-time aspect will only be approached when using the model on new (so-called test) data to detect abnormal situations. In this context, existing deep learning architectures, which use GPU cards, can ensure detecting dangerous situations in real-time.

Furthermore, the system must “learn" to configure itself automatically, taking into account new situations or types of chairlifts not previously recognized. In order to achieve this objective, Mivao includes a phase of studying domain adaptation mechanisms in order to adapt a model that has been previously learned on a given set of chairlifts to a new unseen chairlift.

1.2 Chairlifts datasets

The project dataset is continuously increasing. During this thesis, we used subsets of two versions: 2018 and 2019, further details about the versions will be provided in the following sections. Noteworthy that the 2018 version is composed of images from 21 different chairlifts across different ski resorts, while, in the 2019 version, three new chairlifts were added, making the total number of chairlifts 24. For reasons of confidentiality, we do not give the real names of the chairlifts, we call them hereafter C_0, C_1, C_2, C_3, ..., C_{23}.

1.2.1 Images

Across the ski resorts, there is a wide diversity between the chairlifts: the viewpoint, the background, the carrier 3D geometry, the number of seats, and the camera may be different (see Figure 1.8). A pre-processing is performed on images to have the chairlift carrier roughly in the center. The final images are obtained using the following process:

Figure 1.3 – Examples of variations in lighting conditions during the day in two different chairlifts, namely C_1 (first row) and C_2 (second row).

- For a given chairlift, several video recordings are first made in the ski resort in real conditions, in a 'detection zone' (see Figure 1.1).

- Then, each video is processed to extract a set of 'tracks', each track contains one passage of a single chairlift in front of the camera. Three frames per track are further extracted, respectively, in the beginning, in the middle, and in the end of the passage.

- In addition, each frame is cropped and centered so that the chairlift carrier coarsely at the same 2D position, scale, and orientation. We call the images resulting from this step unitary images (while we refer to the original ones as camera images). Figure 1.4 shows an image before and after transforming it into a unitary image.

Figure 1.4 – Transformation from camera image (left) to unitary image (right), chairlift C_{10}.

1.2.2 Annotations

Annotations provided by BLUECIME

The unitary images are manually labeled 'open', 'closed' or 'between' [3] according to the position of the safety bar of the chairlift carrier in the center of the image, 'between' refers to the cases where the safety bar is not fully open nor closed. The images are also labeled 'empty' or 'occupied' according to the presence of people in the carrier or not. Also, each chairlift comes with information about the geometry of the safety bar. This information is given by two binary masks representing the shape of the safety bar when it is open or closed. Figure 1.8 shows two images from each chairlift and the two corresponding binary masks. We may refer these annotations later as *ground truth*.

Annotations we created

As will be explained later, instance-level bounding box annotations are required to solve the object detection problem. To avoid manual labelling, we have created these annotations based on (i) BLUECIME estimations of the position of the safety bar using SIVAO non-learning-based methods, (ii) the template masks of the safety bar of each chairlift. Since we do not have masks representing the case 'between' we cannot create bounding box annotations for it, that is why we ignore all the images of this class. Then for each unitary image, we compare the ground truth label to SIVAO prediction, if they match, we consider this prediction correct, and we take the coordinates of the estimated position. We ignore all the images where SIVAO prediction does not match the ground truth label. Using the template masks of the chairlift in question, we know the bounding box size, so we set a bounding box based on this position (see Figure 1.5) . This is not equivalent to a ground truth bounding box that would have been annotated by a human, but it gives an approximate position even if it is not 100% accurate.

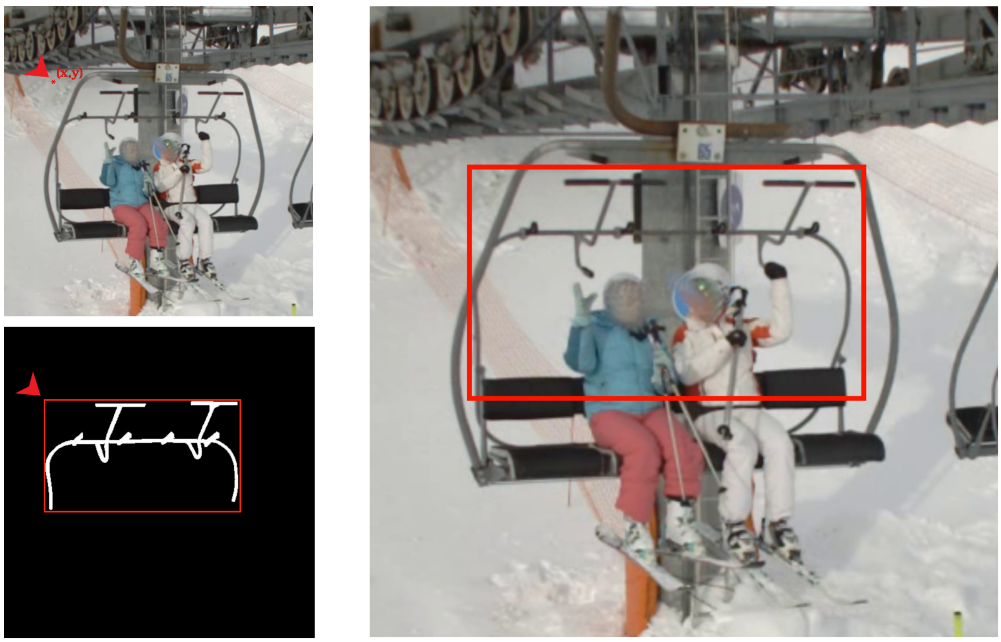


Figure 1.5 – Generate instance-level bounding box annotations for safety bar. (Top left) BLUECIME estimation of the position of the safety bar (x,y) on image from chairlift C_{11}. (Bottom left) the corresponding template mask of the safety bar of chairlift C_{11}. (Right) Instance-level bounding box annotations.

Figure 1.6 shows examples of the instance-level bounding box annotations we have automatically created according to the method explained above. These annotations determine the position of the safety bar with a bounding box (open/closed safety bar).

[3] Open(resp. closed) refers to an open (resp. closed) safety bar, and this is the designation adopted hereafter.

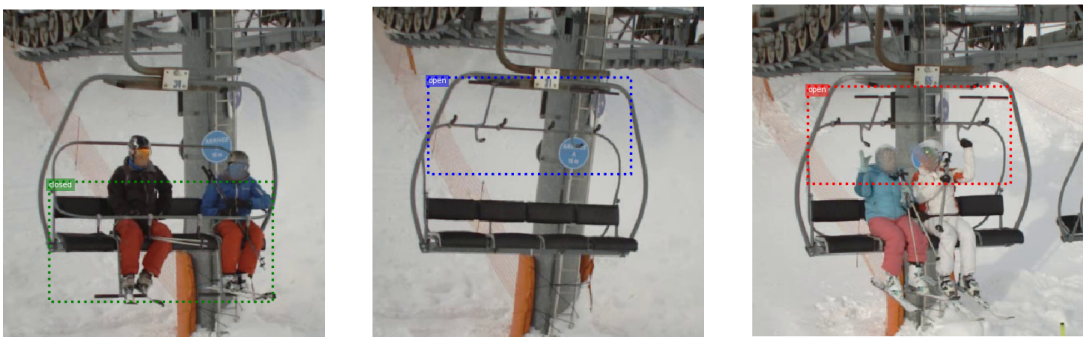


Figure 1.6 – Safety bar instance-level bounding box annotations, examples from chairlift C_{11}.

Annotations provided by an external company

While the project is being developed, instance-level bounding box annotations for person detection have been created by crowdsourcing mediated by an external company independent of the project. These annotations are available for a subset of the 2019 version of the dataset. In this set each image is associated with bounding box annotations for each person with the corresponding label 'adult' or 'child'. Figure 1.7 shows examples of people instance-level bounding box annotations.

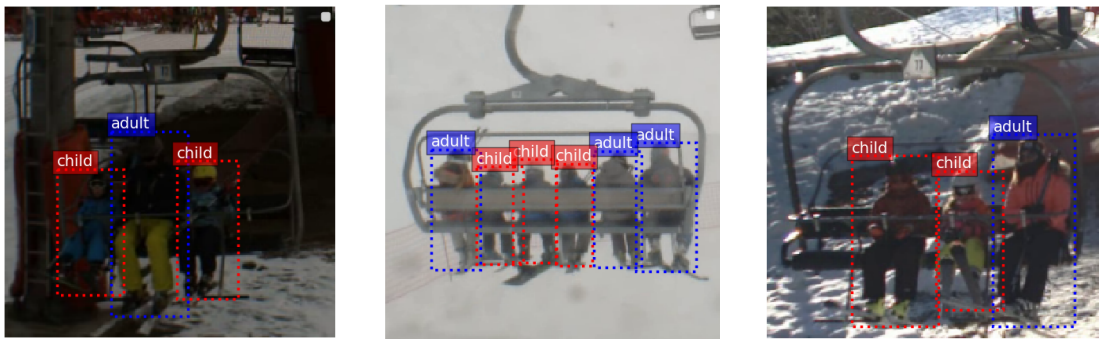


Figure 1.7 – People instance-level bounding box annotations, examples from chairlifts C_8, C_{12} and C_{20}.

1.2.3 Datasets versions

MIVAO 2018 version

Consisting of 21 chairlifts, this version of the dataset has instance-level bounding box annotations for two object categories: 'open' and 'closed'. In addition to image-level ground truth annotation ('occupied' or 'empty', and 'open' or 'closed'). We ignore all the class 'between' images because we do not have their instance-level bounding box annotations, as explained before. Table 1.1 shows the number of images and objects [4] in each class. Table 1.2 presents object distributions for each domain (chairlift). It is clear that the dataset is imbalanced both in the domain distribution and the class distribution. Unsafe situations (*i.e.* occupied carrier with open safety bar) represent a small portion of the total number of objects, only 3%.

In this thesis, experiments will be performed for various difficulty settings, depending on the method we are going to test and the point we attempt to prove or disprove. Therefore, we have created the following datasets with different amounts of training data:

- ***Large chairlifts dataset.*** Which corresponds to the dataset containing all available annotated images. We divided the images of each chairlift into a training set (two-thirds of the images) and a test set (one-third of the images).

[4] The number of images equal to the number of objects because the image contains only one chairlift, and its safety bar is either open or closed and never both.

Table 1.1 – Mivao 2018 dataset, the number of images for each class.

# images = # objects	open		closed	
	occupied	empty	occupied	empty
66254	1667 (3%)	36849(55%)	26401(40%)	1337(2%)
	38516		27738	

Table 1.2 – Mivao 2018 dataset, the number of images for each class, detailed for each chairlift. Last two lines represent the average and standard deviation.

Chairlift	images	open		closed	
		occupied	empty	occupied	empty
C_0	2003	5 (0.3%)	47 (02.4%)	631 (31.5%)	1320(66%)
C_1	4245	146 (3.4%)	1889 (44.5%)	2210 (52.1%)	0(0.0%)
C_2	3163	65 (2.1%)	1746 (55.2%)	1352 (42.7%)	0(0.0%)
C_3	2406	17 (0.7%)	2127 (88.4%)	262 (10.9%)	0(0.0%)
C_4	1468	30 (2.0%)	730 (49.7%)	708 (48.2%)	0(0.0%)
C_5	2689	43 (1.6%)	1724 (64.1%)	920 (34.2%)	2(0.1%)
C_6	2614	110 (4.2%)	1197 (45.8%)	1307 (50.0%)	0(0.0%)
C_7	3823	54 (1.4%)	3413 (89.3%)	356 (09.3%)	0(0.0%)
C_8	4305	119 (2.8%)	2833 (65.8%)	1353 (31.4%)	0(0.0%)
C_9	3425	183 (5.4%)	1762 (51.5%)	1480 (43.2%)	0(0.0%)
C_{10}	5913	197 (3.3%)	4024 (68.1%)	1692 (28.6%)	0(0.0%)
C_{11}	1222	15 (1.2%)	383 (31.3%)	824 (67.4%)	0(0.0%)
C_{12}	912	0 (0.0%)	546 (59.9%)	366 (40.1%)	0(0.0%)
C_{13}	1843	11 (0.6%)	1041 (56.5%)	782 (42.4%)	9(0.5%)
C_{14}	3562	205 (5.8%)	2249 (63.1%)	1108 (31.1%)	0(0.0%)
C_{15}	6393	47 (0.7%)	3424 (53.6%)	2916 (45.6%)	6(0.1%)
C_{16}	3269	192 (5.9%)	1477 (45.2%)	1600 (48.9%)	0(0.0%)
C_{17}	1313	66 (5.0%)	1013 (77.2%)	234 (17.8%)	0(0.0%)
C_{18}	2494	11 (0.4%)	1374 (55.1%)	1109 (44.5%)	0(0.0%)
C_{19}	2296	17 (0.7%)	276 (12.0%)	2003 (87.2%)	0(0.0%)
C_{20}	6896	134 (1.9%)	3574 (51.8%)	3188 (46.2%)	0(0.0%)
Avg.	3154.95	79.38	1754.71	1257.19	63.67
s.d.	1667.26	71.39	1154.31	813.31	287.87

- ***Medium chairlifts dataset.*** For each chairlift, we have randomly chosen 500 images from the training sets of the large dataset; the test set remained the same for a fair comparison.
- ***Small chairlifts dataset.*** We have randomly chosen 100 training images from the training sets of the large dataset, the test set is kept the same.
- ***Tiny chairlifts dataset.*** We have chosen a smaller number of training images, only 20 images per chairlift. Because the size of the dataset is too small, and there is an imbalance in the original dataset classes, we ensured that there are 10 images per class for each chairlift. The test set remained the same.

Table 1.5 shows the distribution of images between the two classes in the training and test sets of the four datasets.

MIVAO 2019 version

Consisting of 23 chairlifts, this version of the dataset has instance-level bounding box annotations for two object categories: 'adult' and 'child'. In addition to image-level ground truth annotation ('occupied' or 'empty', and 'open ', 'closed' or 'between'). Table 1.3 shows statistical information about the annotated images and objects. We have a great imbalance between 'adult' and 'child' classes, as well as 'open ', 'closed', and 'between' classes. However, the domains are perfectly balanced in terms of the number of images see Table 1.4.

Table 1.3 – Mivao 2019 dataset, the number of images and objects from each class.

# images	# objects	
22792	adult	child
	43288 (89%)	5386 (11%)

Table 1.4 – Mivao 2019 dataset, the number of images and objects from each class, detailed for each chairlift. Last two lines represent the average and standard deviation.

chairlift	images	adult	child	closed	open	between
C_0	992	2442 (85.96%)	399 (14.0%)	952 (95.97%)	21 (2.12%)	19 (1.92%)
C_1	995	2396 (90.86%)	241 (9.14%)	869 (87.34%)	44 (4.42%)	82 (8.24%)
C_2	998	2155 (93.41%)	152 (6.59%)	925 (92.69%)	52 (5.21%)	21 (2.10%)
C_3	991	2038 (84.56%)	372 (15.4%)	670 (67.61%)	39 (3.94%)	282(28.5%)
C_4	994	2324 (95.21%)	117 (4.79%)	957 (96.28%)	28 (2.82%)	9 (0.91%)
C_5	992	2705 (87.17%)	398 (12.8%)	786 (79.23%)	59 (5.95%)	147(14.8%)
C_6	995	2735 (93.15%)	201 (6.85%)	825 (82.91%)	51 (5.13%)	119(12.0%)
C_7	996	1542 (94.02%)	98 (5.98%)	658 (66.06%)	109(10.9%)	229(23.0%)
C_8	995	1980 (92.31%)	165 (7.69%)	617 (62.01%)	126(12.7%)	252(25.3%)
C_9	999	2905 (95.43%)	139 (4.57%)	819 (81.98%)	94 (9.41%)	86 (8.61%)
C_{10}	999	2459 (91.28%)	235 (8.72%)	666 (66.67%)	101(10.1%)	232(23.2%)
C_{11}	999	2742 (96.82%)	90 (3.18%)	955 (95.60%)	1 (0.10%)	43 (4.30%)
C_{12}	989	2903 (89.24%)	350 (10.8%)	898 (90.80%)	30 (3.03%)	61 (6.17%)
C_{13}	989	3018 (88.04%)	410 (12.0%)	852 (86.15%)	7 (0.71%)	130(13.1%)
C_{14}	992	3168 (93.98%)	203 (6.02%)	915 (92.24%)	32 (3.23%)	45 (4.54%)
C_{15}	980	2657 (87.06%)	395 (12.9%)	905 (92.35%)	20 (2.04%)	55 (5.61%)
C_{17}	991	2368 (95.68%)	107 (4.32%)	816 (82.34%)	61 (6.16%)	114(11.5%)
C_{18}	993	2333 (88.91%)	291 (11.1%)	819 (82.48%)	36 (3.63%)	138(13.9%)
C_{19}	991	4038 (92.23%)	340 (7.77%)	964 (97.28%)	1 (0.10%)	26 (2.62%)
C_{20}	969	2238 (91.68%)	203 (8.32%)	846 (87.31%)	36 (3.72%)	87 (8.98%)
C_{21}	979	2788 (92.20%)	236 (7.80%)	777 (79.37%)	79 (8.07%)	123(12.6%)
C_{22}	977	2704 (95.21%)	136 (4.79%)	903 (92.43%)	20 (2.05%)	54 (5.53%)
C_{23}	999	2453 (95.78%)	108 (4.22%)	725 (72.57%)	73 (7.31%)	201(20.1%)
Avg.	991.04	2569.17	234.17	831.26	48.70	111.09
s.d.	7.82	490.13	112.03	104.95	34.63	80.48

1.3 Objectives

In this thesis, we aim at identifying unsafe situations which refer to images where:

- the chairlift carries passengers, and the safety bar is not completely closed.

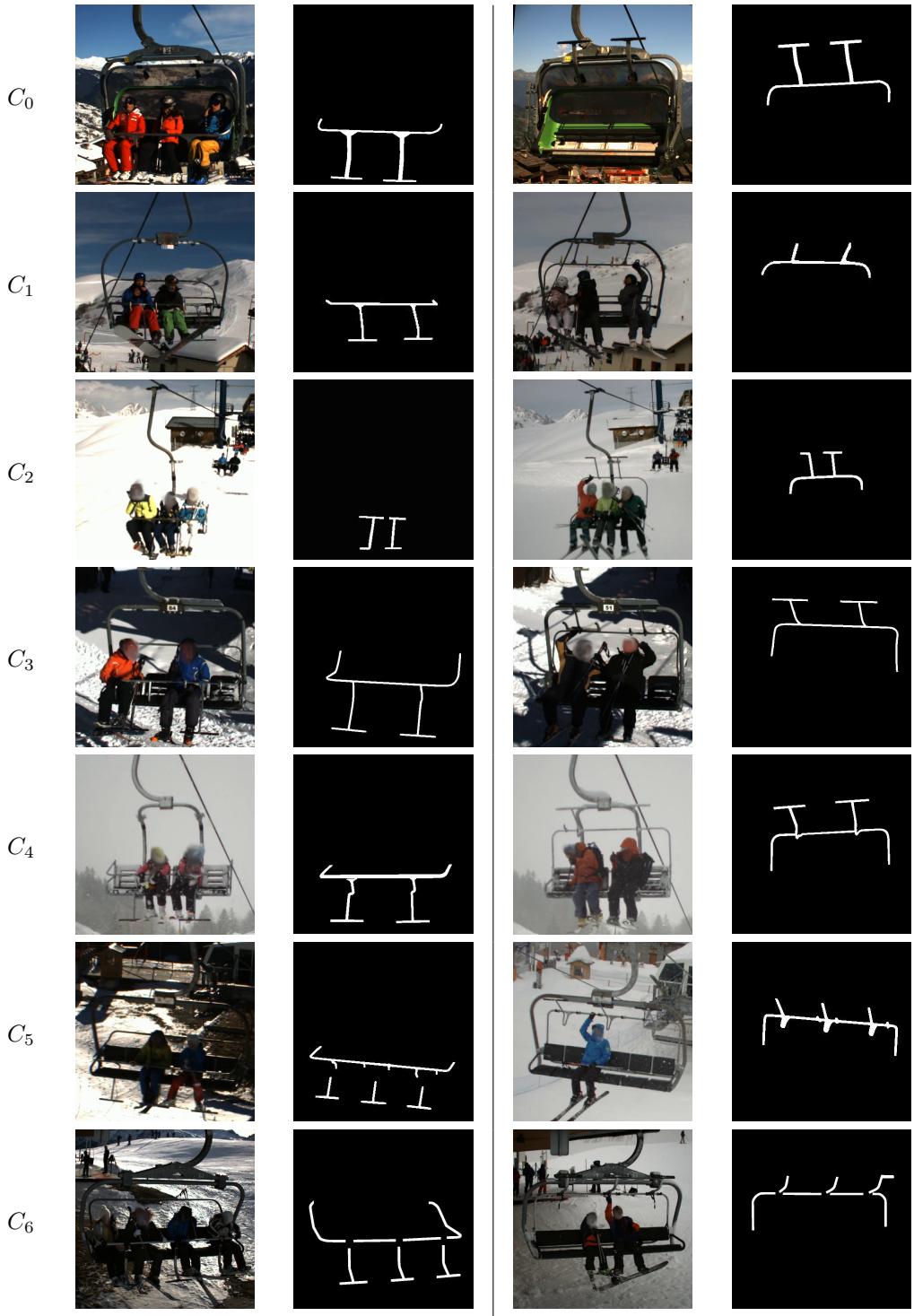
On the contrary, safe situations are images where:

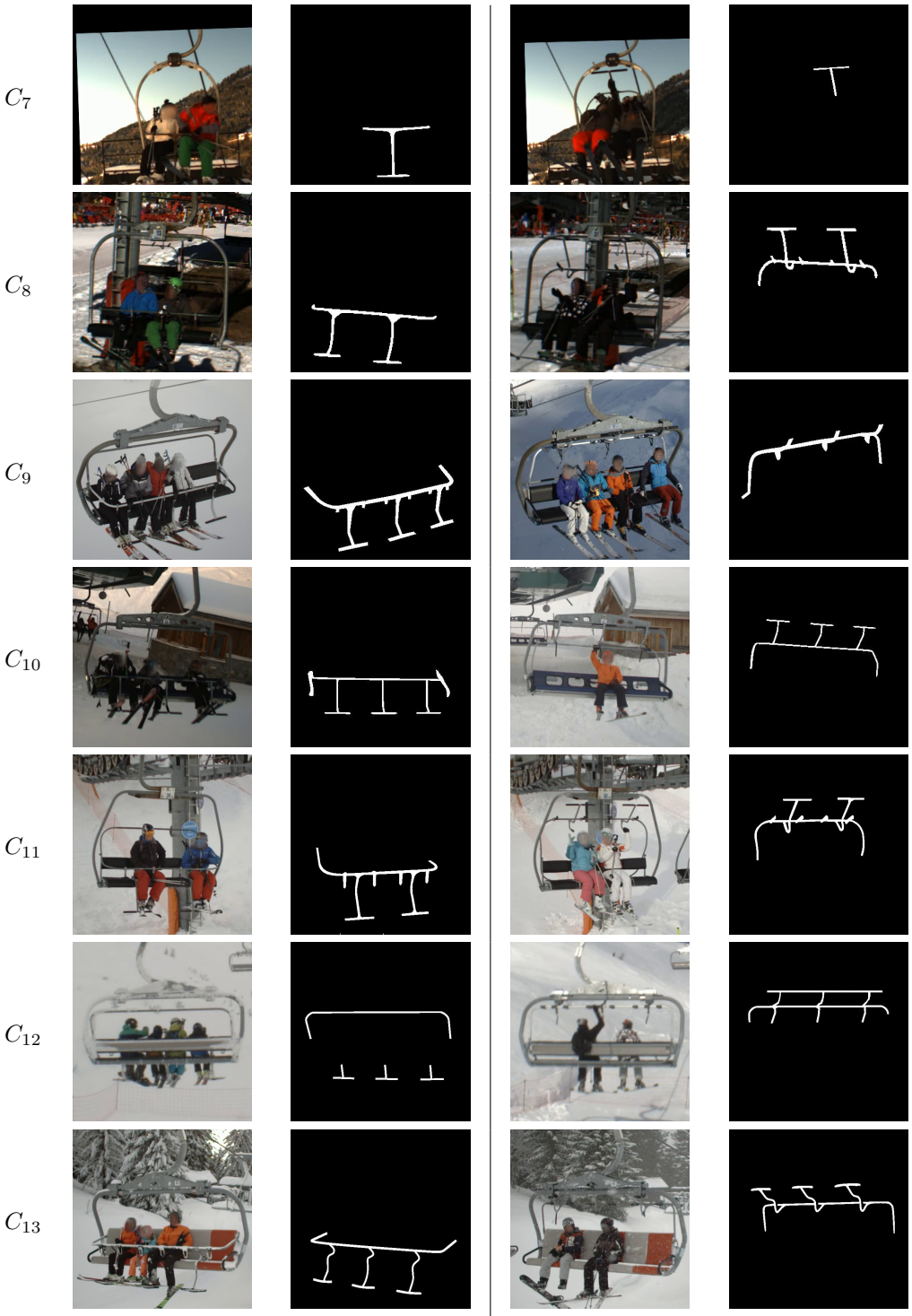
- the chairlift carries passengers and the safety bar is completely closed;
- the chairlift does not carry any passengers, the safety bar is by default open in these cases (except for a few exceptions).

Children could be considered unsafe if they are not accompanied by adults. Therefore, we are keen on detecting people in the images and classifying them into adults and children, in addition to localize the safety bar to determine if it is open or closed. Chapter 3 is dedicated to showing the results of a state of the art object detector for this problem.

Table 1.5 – Distribution of the images between the two classes (open and closed) in training and test sets of the *large, medium, small* and *tiny chairlifts datasets*. Note that the same test set is used for all the experiments.

Chairlift	Training								Test	
	large		medium		small		tiny			
	open	closed	open	closed	open	closed	open	closed	open	closed
C_0	40	1566	15	485	1	99	10	10	12	385
C_1	1597	1748	243	257	61	39	10	10	438	462
C_2	1403	1075	295	205	63	37	10	10	408	277
C_3	1700	202	445	55	91	9	10	10	444	60
C_4	609	560	255	245	54	46	10	10	151	148
C_5	1405	714	334	166	65	35	10	10	362	208
C_6	1024	1005	261	239	51	49	10	10	283	302
C_7	2745	267	447	53	82	18	10	10	722	89
C_8	2322	1060	333	167	63	37	10	10	630	293
C_9	1556	1151	265	235	54	46	10	10	389	329
C_{10}	3362	1348	365	135	74	26	10	10	859	344
C_{11}	305	658	165	335	29	71	10	10	93	166
C_{12}	421	305	288	212	53	47	10	10	125	62
C_{13}	851	607	289	211	55	45	10	10	201	184
C_{14}	1902	850	327	173	63	37	10	10	551	258
C_{15}	2708	2294	295	205	64	36	10	10	763	628
C_{16}	1364	1217	265	235	59	41	10	10	305	383
C_{17}	858	186	410	90	84	16	10	10	221	48
C_{18}	1087	887	282	218	64	36	10	10	298	222
C_{19}	226	1581	70	430	12	88	10	10	67	422
C_{20}	2837	2172	302	198	61	39	10	10	813	847
Total	51774		10500		2100		420		14252	


C_0
C_1
C_2
C_3
C_4
C_5
C_6
closed safety bar image
closed safety bar mask
open safety bar image
open safety bar mask


C_7
C_8
C_9
C_{10}
C_{11}
C_{12}
C_{13}
closed safety bar image
closed safety bar mask
open safety bar image
open safety bar mask

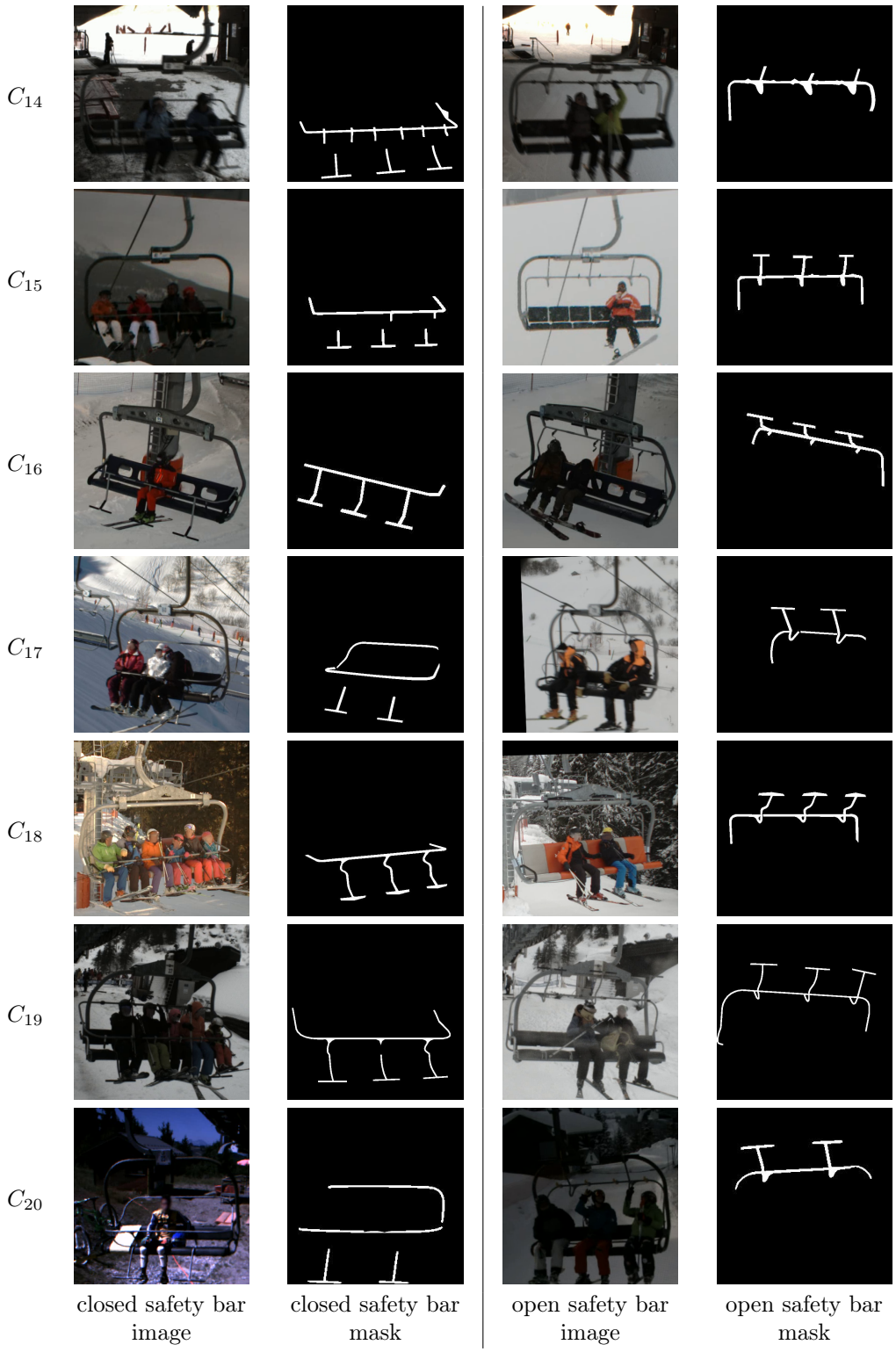


Figure 1.8 – Example images from 21 different chairlifts with the corresponding masks. On the left images and masks with 'closed' safety bar, on the right images and masks with 'open' safety bar which represent a dangerous situation must be reported (Mivao 2018 dataset).

Chapter 2

State of the Art

In this thesis, our objective is to propose methods based on deep learning to identify dangerous situations in chairlifts. These methods are intended to improve the performance of the existing system Sivao by enhancing the accuracy and reducing the time needed to configure the system for a new chairlift. This chapter presents an overview of the available literature on the principal aspects related to our objectives. To begin with, we review the solutions that already exist in the field of chairlift safety. Second, we study image classification methods based on deep learning and highlight their potential limitations in our context. Third, we present object detection and try to explain its main advantages for the chairlift safety problem. The fourth important aspect to consider is domain adaptation, *i.e.* the ability to transfer a previously learned model from a source domain to a target domain, which in our case may be a new chairlift. Finally, in the chairlift safety problem, we have at our disposal a knowledge of both the particular geometry of each chairlift and the common information shared by all chairlifts. Therefore, it is worth going over what has already been proposed to exploit this information in deep learning models.

2.1 Background of the chairlift safety problem

Safety can be defined in different terms depending on the context *e.g.* keeping the seat belt fastened during take-off and landing in a flight, protecting against robbery in a crowded place, preventing the risk of explosion in a factory, physical distancing between people in the time of the COVID-19 epidemic, *etc.* In the chairlift area of the ski resorts, users' safety is a matter of great importance, and the term safety covers mitigating any abnormal situation *i.e.* an open safety bar after leaving the boarding station, a person slipping from his carrier, an unaccompanied child, an overloaded carrier, collapses which injures the person who falls and creates an obstacle for others, *etc.* The anomalous or abnormal activity analysis in chairlifts is complicated due to several real-world constraints including but not limited to: variability in weather conditions, variability in lighting conditions during the day, variability in shape and appearance of the chairlifts.

A number of studies have been conducted to analyze the causes and patterns of skiing injuries and their relation to many factors *e.g.* fatigue [5], skier experience [6], skiing errors or speed [7], infrastructure and quality of equipment [8], environmental conditions [9], *etc.* The ultimate goal of such studies is to improve skiers' safety by reducing the chances of injury. Based on the region, these studies collect data in different ways, such as interviews with skiers, questionnaire surveys, analysis of ski lift usage, reviewing patient records in mountainside clinics, *etc.* Some studies like [10] and [11] applied machine-learning techniques over skiing-related data to predict the risk of injuries. None of these studies focused on the same task as this thesis, and the problems they address and the approaches they use are not relevant to our work.

In this thesis, we tackle the problem of monitoring chairlifts as a transport means for ski or other mountain sports like hiking or cycling. We focus on the boarding station since this is a potentially high-risk area, given the density of skiers and the mechanical nature of chairlifts. We study methods to improve the safety of the chairlift user by monitoring the station, identifying developing problem situations, and triggering an alarm to allow the operator to undertake appropriate corrective actions. The majority of the chairlifts have one or more human operators

monitoring the boarding station, and generally these operators will react to abnormal situations, by slowing down or stopping the chairlift. However, the operators are not always alert or aware of emergencies and may be out of position to act immediately when necessary. Besides, continuous manual monitoring for long duration is difficult for humans because it is laborious and time-consuming. However, maximum accuracy and minimum response time are expected in such a real-world application. Automatic video surveillance analysis is a solution to tedious human tasks, must be highly valid, accurate, and reliable while reducing maintenance costs and decreasing system response time.

2.1.1 Video surveillance systems

Over the last years, video surveillance systems have developed from simple video record-and-display systems to automatic intelligent systems. Thanks to the development of sensing devices, storage units, wireless broadband technologies, high-definition cameras, as well as data processing and analysis [12]. The survey by Joshi and Thakore [13], shows that there are three types of video surveillance systems, based on their degree of dependence on the human operator: (i) manual, (ii) semi-autonomous, and (iii) fully autonomous. In (i), a human operator is in charge of the monitoring and analyzing the video content, and making the decision. This type of system is considered passive, and it is the oldest type, but it is still in use today. In (ii) , automatic video processing algorithms substitute for the operator in certain tasks, while the rest of the work is performed by the human. In (iii), the system is responsible for detecting, analyzing and anticipating events or behaviors of objects, without human intervention.

To develop an understanding of the visual events in the scene, the intelligent visual surveillance (semi or fully automated) involves the analysis and interpretation of object behaviours, as well as object detection and tracking [14]. This process involves many different disciplines, such as image processing and enhancement techniques, distributed computing infrastructures, pattern recognition, decision-making, and other machine learning algorithms [15]. Research developments in these technologies have raised expectations for automatic surveillance systems, which are now intended to improve accuracy and reduce system response time.

Video surveillance systems have gained great attention as application-oriented research [16], due to their wide range of application areas such as security management [17], elderly care [18], accident detection [19], crowd monitoring [20], factory automation [21], highway visibility detection [22], airport monitoring [23], *etc.* Chairlift safety based on intelligent video systems is a specific area that has been so far little explored in the literature. However the analysis of surveillance videos entails different algorithms from research areas that are relevant to our context *e.g.* classification [24], object detection [25, 26], object tracking [27, 28], action recognition [29], person detection [30], anomaly detection [31, 32], *etc.*

2.1.2 Systems and methods to improve chairlift safety

Concerning the safety problem of chairlifts, there has been little work proposed in the literature to address the problem of monitoring chairlifts as a means of transport compared to autonomous vehicles. For example. EVEREST project [1] aims to estimate the capacity of systems based on image processing and analysis, to provide technical support and surveillance functions for guided transport in the mountains, allowing industries and academics to propose algorithms for the detection of risk situations. To achieve this objective, a 96-hour video database filmed in three French ski resorts, and Ra *et al.* developed a semi-automatic tool to annotate this database and anonymize passengers using face tracking approach [33]. The annotated dataset is acquired from different points of view on the boarding/disembarking stations of the ski lifts, and presents three dangerous situations caused by inappropriate user behavior: (i) presence of passenger beyond the ski lift arrival area, (ii) incorrect position of the passenger with respect to the ski lift departure area, (iii) the safety bar is not fully closed beyond a given area with the presence of the passenger on the seat. Because these situations are rare, some of the samples are played by actors. EVEREST project involves a challenge to solve the problem of detecting potentially dangerous situations. The algorithms proposed by the candidates will be evaluated during a workshop specially dedicated to the study, which will take place in 2021.

[1] https://everest.ifsttar.fr/

U.S. Patent No. 10,628,679 (2020) [1] extensively describes systems and methods to improve the operation of chairlifts to enhance safety for users at the boarding and disembarking stations. The workflow of such a system is illustrated in Figure 2.1. Boarding and disembarking stations are monitored using video cameras (102) that transmits live video to a video processing module (224) results in image sequences. Then an artificial intelligence (AI) engine (226) identifies developing problem situations, and the inference processing module (228) takes appropriate corrective action by sending an alert (110) or by interacting with the lift motor controller (106) to slow down or stop the lift.

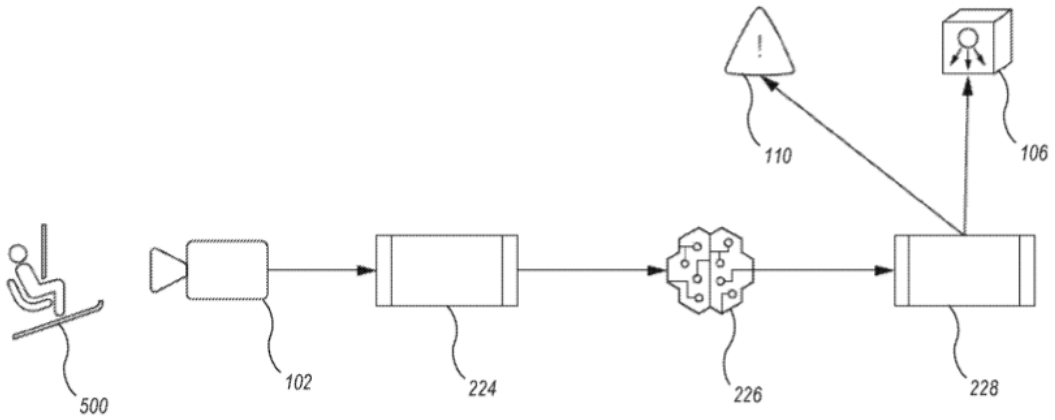


Figure 2.1 – A block diagram demonstrating the workflow of an example system to improve lift operations (Figure from [1]). Numbers indications from left to right 500: lift rider, 102: video camera, 224: video processing module, 226: artificial intelligence engine, 228: inference processing module, 110: ski lift alert system, 106: ski lift motor controller.

According to this patent, the AI engine of the ski lift problem detection system must be trained to classify situations into normal, a developing problem, a minor problem, or a major problem. Many realization modes have been shown in the patent, for example, by adding long short-term memory (LSTM) to help the AI engine to have contextual awareness, or by using a convolutional neural network and training it by back-propagation and using techniques such as data augmentation, fine-tuning, *etc*.

Sivao (described in Chapter 1) is a kind of these systems with some differences. Sivao has a camera only in the boarding station. The analysis part mainly includes analysis of frames extracted from videos using image processing techniques to make a decision about potential problems. The system does not initiate any action to control the chairlift except to trigger the alarm. Sivao serves as a backup monitoring system alongside human monitoring. The objective of Mivao project (also described in Chapter 1) is to replace the deployed hand-tuned image processing techniques by deep learning. As part of Mivao project, Bascol *et al*. [34] proposed to use ResNet with some domain adaptation components, resulting in an end-to-end model improving the performance of Sivao in classifying safe and unsafe situations. In [35], Bascol *et al*. propose a multisource domain adaptation method that selects and weights the sources based on inter-domain distances. They applied their method on chairlift safety problem considering each chairlift as domain. Also within the context of Mivao, Muzeau *et al*. [36] proposed a classification tool to classify empty chairlifts from those with passengers on board, using linear discriminant analysis, which is a dimensionality reduction alongside classification technique.

In this thesis, at first, we decided to approach the problem as an object detection task. We performed preliminary tests using an advanced object detector to detect the safety bar position and the people. The results of the object detection will then be beneficial and easy to use in decision making in different scenarios *i.e*. a situation will be considered dangerous if people are detected in the chairlift when the safety bar is not closed correctly, or a child is detected in the chairlift without being accompanied by an adult, *etc*. Then we investigated the problem from a domain adaptive perspective to assess the ability of object detector trained on a chairlift to

perform well on new unseen chairlift [37]. While in [38] we opted to exploit the geometrical constraint by guiding shallow and deep classifiers using binary masks representing the safety bar, then we proposed a solution to boost the generalizability of the deeper classifier. In the later sections of this chapter, we present an overview of the available literature on each of these aspects.

2.2 Image classification

One of the most critical factors in the chairlift safety problem is the position of the safety bar. A natural approach to solve this problem is to use image classification, where we classify the images into the categories: open safety bar, closed safety bar. This section reviews the image classification problem and how methods based on Convolutional Neural Networks (CNN) can be applied in our context and what their potential limitations are. First, as we focus on CNN-based methods, we will briefly introduce CNN. Next, we present an image classification pipeline, survey the main CNN architectures conventionally employed for image classification, and finally, underline the limitations of CNN-based image classification in the chairlift safety problem.

Image classification refers to the task of taking a single image, identifying the features it contains in terms of objects, thereafter classifying the image according to the dominant object into one of a finite number of classes [39]. Image classification is a major problem in computer vision which, despite its straightforward nature, has a wide variety of practical applications [40]. In addition, many other supposedly distinct computer vision tasks (such as object detection, or image segmentation) can be reduced to image classification [41]. A good image classification model should be invariant to the combination of all extra-class variations, while at the same time conserving sensitivity to inter-class variations [42].

Deep learning (DL) provides high performance and flexibility in many fields *e.g.* computer vision, speech recognition, natural language processing, *etc.* In the following, we present CNN, the DL technique that recently achieved favorable results in recognition and detection tasks.

2.2.1 A brief introduction to Convolutional Neural Networks

Before introducing any type of neural networks (NNs), it is essential to introduce Perceptron [43], the linear binary classifier inspired by the biological neurons. Perceptron is a mathematical function [44] (illustrated in Figure 2.2a), that takes inputs X, weights them by a set of weights W, sums them up, and passes this weighted sum plus a bias b through an activation function $\varphi(.)$ as follows:

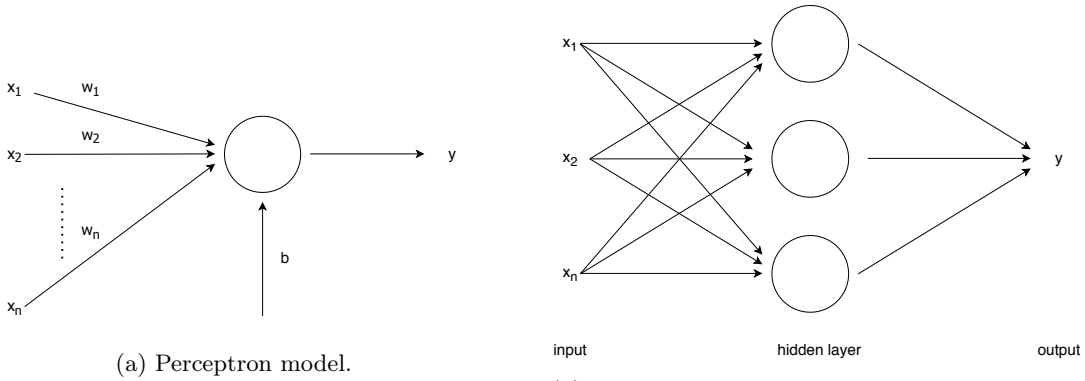
$$\hat{y} = \varphi(W^T X + b) \tag{2.1}$$

The perceptron learns the value of the parameters θ *i.e.* weights W and biases b, that lead to the best approximation of the function that maps each input $x \in X$ to its category $y \in Y$, *i.e.* $y = \varphi^*(x)$. Stacking perceptrons in layers results in multilayer perceptrons (MLPs) or feedforward neural networks [43]. Mathematically, each layer takes as input the output of the layer before it [44]:

$$f(X) = \varphi(W_n^T \varphi(W_{n-1}^T \dots \varphi(W_1^T X + b_1) + b_{n-1}) + b_n) \tag{2.2}$$

Where, $W_{1:n}, b_{1:n}$ are the weights and bias of layers $1 \dots n$, and the activation function $\varphi(.)$ is a non-linear function like sigmoid, tanh or rectified linear unit (ReLU). MLPs take an input and map it to an output through a series of hidden layers connected to each other in a fully-connected manner. Each hidden layer consists of a set of neurons completely independent and does not share any parameters. This complete connectivity between adjacent layers, and the absence of connections within a single layer leads to a large number of parameters, that could lead then to over-fitting. Figure 2.2b illustrates an MLP with one hidden layer. When a neural network has multiple hidden layers, it is generally considered to be a 'deep' neural network [44].

Convolutional Neural Networks [45] (CNNs) are a kind of NNs that use convolution instead of general matrix multiplication in one or more of their layers. CNNs are dedicated for grid-like data such as sentences which can be regarded as a 1-D grid of words, or images which can be regarded as a 2-D grid of pixels [46]. CNNs achieved outstanding performance in a wide range



(a) Perceptron model.

(b) Feed forward neural network with only one hidden layer.

Figure 2.2 – Perceptron and multilayer perceptrons.

of problems, based on an end-to-end pipeline in which all stages can be trained altogether by optimizing a single objective function. The key innovation of CNNs is that the layers are not fully connected, reducing the number of parameters to learn. Instead, the neurons in a layer are only connected to a small region of the previous layer. CNNs generally consist of three types of layers: convolutional, pooling, and fully connected.

1. Convolutional layers: the principal objective of the convolutional layer is to extract features from the input and produce feature maps. The convolutional layers consist of a set of learnable filters. A filter convolves the entire input (network input or intermediate feature map) to produce a feature map, each member of this filter is used at every position of the input by connecting locally to a small region at a time [47], this local connectivity maintains rich spatial information [48]. Formally, the non-linear activation function $\varphi(.)$ is used to compute the feature value at location (i, j) of the k^{th} feature map y_k, by computing the dot product between w_k the weights of the k^{th} filter, and $x_{i,j}$ a small region of the input data, and then adding a bias term b_k [49]:

$$y_{i,j,k} = \varphi({w_k}^T * x_{i,j} + b_k) \tag{2.3}$$

   CNNs have sparse connectivity due to the fact that the filter is smaller than the input [46]. That is to say, each neuron in a layer receives inputs from a set of neurons located in a restricted area in the preceding layer (see Figure 2.3). The region of the input that affects the activation of a feature is called the receptive field (field of view of a neuron). When the number of connections of each layer is limited, then fewer parameters are needed, leading to a reduction of the computational load during training [50].

2. Pooling layers: the aim of pooling is to reduce the number of parameters to be learned in the following layers, it is an effective way to increase the size of the receptive field quickly, and it helps to decrease the overfitting [51]. Reducing the dimensionality of each feature map also reduces the sensitivity of the output to shift transformations, and retains only the most important information. Spatial pooling can be of different types: max, average, sum, *etc.* Boureau *et al.* [52] demonstrated the theoretical details about the performances of average and max pooling.

3. Fully-connected (FC) layers: in FC-layers, neurons are fully pairwise connected to the neurons in adjacent layers. Whereas, neurons within a single layer share no connections. Spatial information is lost because the FC layer transforms the feature maps into a feature vector representing the entire image [53]. 90% of CNN parameters are in its FC layers, thus, recent models trend towards removing the last FC layers and sometimes replacing them with a global average pooling layer [50, 2], in order to have fewer parameters and more accurate classification. However, Zhang *et al.* proposed a study [54] that emphasizes the importance of FC layers in the transfer of CNN visual representations, as they significantly influence achieving high accuracy in the target domains by fine-tuning the pre-trained model.

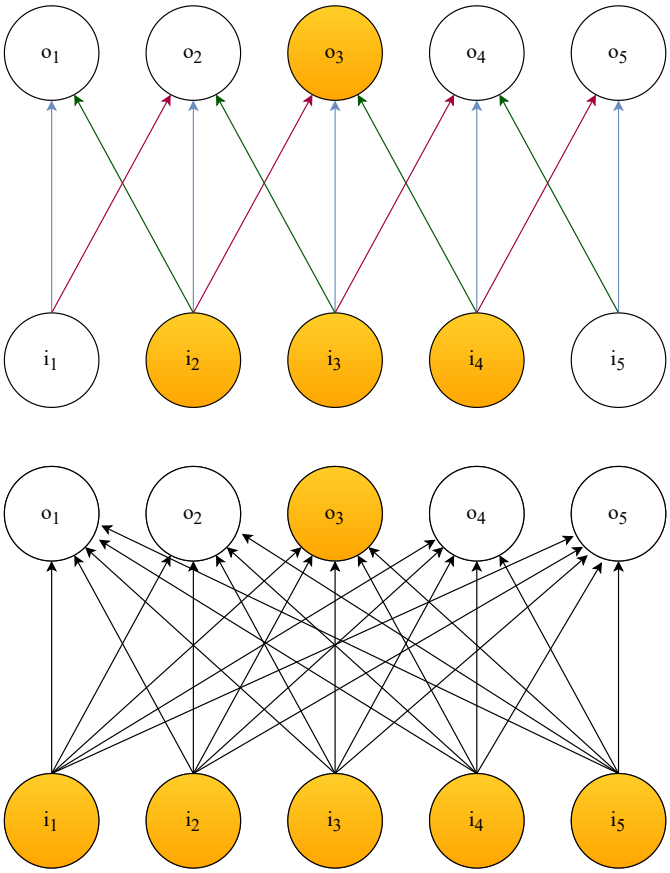


Figure 2.3 – Sparse connectivity and weight sharing. Convolution (top): the green, blue and red arrows indicate uses of the first, second and third (respectively) element of a 3-element kernel, each element is used at all inputs *i.e.* weight sharing. Because the kernel size is 3, each output unit is affected by only 3 input units. For example, the highlighted input units i_2, i_3 and i_4 affect the highlighted output unit o_3, and this is the receptive filed of o_3 *i.e.* sparse connectivity. Fully connected (bottom): each black arrow indicates an element of the weight matrix, note that each arrow is used only once *i.e.* no weights sharing. Each output unit is affected by all the inputs. For example, o_3 is affected by i_1, i_2, i_3, i_4 and i_5 *i.e.* no sparse connectivity.

CNNs use learning algorithms to find the best fitting set of parameters (*i.e.* weights and biases) to achieve the expected network output [47]. The most commonly used algorithm for this purpose is back-propagation [45]. Back-propagation computes the gradient of an objective (loss) function with respect to the network parameters, to find out how to adjust those parameters to minimize the errors that reduce performance.

2.2.2 CNN-based image classification

Compared with other methods, CNNs achieved better classification accuracy on large scale datasets because of their ability to jointly learn the features and the classifier [55]. Since the success of AlexNet, the winner of ILSVRC 2012, major advances were made in classification accuracy by either reducing filter size or expanding the network depth.

The complete pipeline of a CNN-based image classifier can be formalized as follows. A set of N images, each image x_n labeled with one of K different classes y_n, this data is referred to as the training set. The training set is used to learn an approximate function mapping the input image to the output label $y_n = f^*(x_n)$, this step is referred to as training a classifier, or learning a model. Training CNN is done by minimizing a loss function [53]:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \ell(\theta; y_n, o_n) \tag{2.4}$$

Where, θ the parameter of the CNN, o_n is the output of the CNN giving the input x_n:

$$o_n = f(\theta, x_n) \tag{2.5}$$

Training CNN-based image classification network by back-propagation can be divided into two stages [49]:

- Forward stage: first depending on the weights and bias for each layer a label o_n is predicted for the input image x_n *i.e.* Equation 2.5. The loss cost is calculated by using the predicted output o_n and the real output y_n *i.e.* Equation 2.4.
- Backward stage: depending on the loss cost, the gradients are computed for each parameter. Once calculated, these gradients are propagated through all the layers, from the output at the top (where the network produces its prediction o_n) to the bottom (where the external input x_n is fed). Using the gradients, the parameters are updated for the next iteration.

After a sufficient number of iterations, the quality of the model is assessed by having it predict the labels of a new set of images that it has never seen before. Then, the actual labels (called ground truth) of these images will be compared to those predicted by the classifier. Intuitively, many of the predictions are expected to match the true answers. Training the classifier could be done in several ways:

- Training the CNN from scratch *i.e.* all the network parameters θ are randomly initialized. This way requires a large amount of training data; otherwise, over-fitting may occur and the model will not generalize to new data.
- Using the features of a pre-trained network and train an independent classifier that can be based on a neural network or not (*i.e.* SVM). Such a network would have already learned features that are useful for most computer vision problems, and leveraging such features would allow to reach a better accuracy than any method that would only rely on the available data. However, in some application context those features could be so general or irrelevant, *e.g.* a classifier trained to classify dogs and cats is not expected to perform well in classifying safe and unsafe situations in a chairlift.
- Fine-tuning a pre-trained network by training only the top layers of the network (usually the fully connected layers), more details about fine-tuning will be covered in Section 2.4.

2.2.3 Popular CNN architectures

The most popular CNN architectures are the winners of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [56], we present them below.

- AlexNet [57]: ILSVRC 2012 winner of both the classification and localization was the SuperVision team, their network AlexNet achieved a top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. Based on LeNet [58], combined with data augmentation, Rectified Linear Units (ReLUs) as an alternative to the traditional sigmoid activation function, dropout for reducing over-fitting, and GPU implementation. It proved the effectiveness of its distinguished resurgence and opened a new area for computer vision. The network is composed of 5 convolution layers followed by 3 fully connected layers.
- ZFNet [59]: ILSVRC 2013 winner for image classification was Clarifai, with 14.7% as top-5 validation error rate. Their network ZFNet is essentially a refined AlexNet with a small adjustment: use 7×7 filters instead of 11×11 filters in the first convolution layer to maintain more data. The intuition behind this is that larger filters lead to losing a lot of pixel information, which could be retained by having smaller filter sizes in the earlier convolution layers. Another important difference is that the sparse connections used in AlexNet's layers 3,4,5 are replaced with dense connections. ZFNet is composed of 8 layers in total.
- VGGNet [60]: ILSVRC 2014 runner-up, with 7.3% error rate, although it was not the winner, VGG is up to this time one of the most popular CNN architectures due to its clarity and effectiveness. The main idea was to replace large-kernel convolution filters by

stacking several small-kernel filters. It precisely uses 3×3 kernel-sized filters with stride and padding of 1, along with 2×2 max-pooling layers with stride 2, based on the idea: multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increase the depth of the network which enables it to learn more complex features at a lower cost. The network is composed of 16 convolutional layers and 3 fully connected layers and known as VGG16; another version with 3 more convolutional layers was proposed and named VGG19.

- InceptionNet [50]: ILSVRC 2014 winner was GoogLeNet team with a top 5 error rate of 6.7%. Instead of the classical stacking up of convolution and max-pooling layer consecutively, it stacks up Inception modules, consisting of multiple parallel convolutions and max-pooling layers with different kernel sizes, to capture details at varied scales. It uses 1×1 convolutional layer (network in network idea) to reduce the depth of feature volume output. There are currently 4 InceptionNet versions. InceptionNet v1 has 9 inception modules stacked linearly, each of them consists of 3 parallel convolutional layers, consequently the network is 27 layers deep.

- ResNet [2]: ILSVRC 2015 winner with 3.57% error rate. Although the deeper the network, the better the performance, if more layers are added, the accuracy will begin to saturate at some point and eventually degrade. Hence, instead of transforming the input representation to output representation, ResNet sequentially stacks residual blocks (Figure 2.4), each of them computes the change (residual) between its output $H(x)$ and its input x *i.e.* $F(x) = H(x) - x$, and adds that to its input to produce its output representation *i.e.* $H(x) = F(x) + x$. Residual mapping is easier to learn and inputs can propagate faster through the residual connections between layers. On ILSVRC ResNet had a depth of 152 layers, 8 times deeper than VGGNet. However many versions exist today with a different depth such as ResNet50, ResNet101, *etc.*

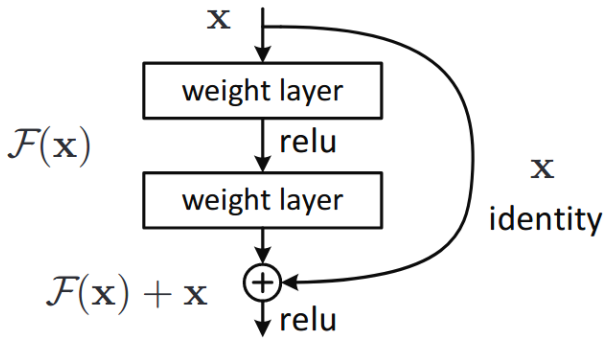


Figure 2.4 – Residual block (Figure from [2]).

2.2.4 Limitations in the chairlift safety problem

Deep learning methods for image classification have a great advantage: models learn only from examples, eliminating the need for feature engineering. With sufficient annotated examples available, these methods have proven to be very accurate. In our context, each chairlift has its own unique elements that need to be identified by the system *e.g.* chairlift design, safety bar shape, *etc.*, therefore a sufficient number of annotated examples is required per chairlift to train the system. By learning a general model on various chairlifts, we may lose chairlift-specific features when these features are not well represented in the training data. Furthermore, there is no guarantee that the model will generalize well to a previously unseen chairlift with a particular geometry. In most applications, including ours, deep learning can be seen as a black box, where internal network information is usually not very clear. We do not exactly know which element of the scene is behind the decision made by the system. This makes it harder to analyze the performance of the network, and to detect a potential problem in the system architecture. In the chairlift safety problem, the object of interest *i.e.* safety bar, is a relatively small part of a complex scene, making its signature embedded in the background noise such as clutter, environmental conditions, geometric variations. Under these adverse conditions, a CNN-based

image classification system is far from ideal for performance and reliability. Therefore, in the next section, we will consider *object detention* as a potential solution to this problem.

2.3 Object detection

Unlike image classification, in object detection we explicitly train a model to detect specific elements of the scene. This is particularly useful in chairlift safety because it enables the accurate detection of visual information related to safety *i.e.* safety bar, people, skis, *etc.* This section analyzes the different deep learning-based methods for object detection and studies how they can be applied to our problem.

To develop a comprehensive understanding of visual data and draw high-level information from images, we must bridge the gap between the raw pixel intensities and the semantic information contained in images. This requires an approach that goes beyond image classification. Object detection -as a natural extension of image classification- is a stepping stone on the path to solve this paradigm.

Object detection consists of precisely locating all the instances of predefined object categories in the image and classifying them correctly. The location is usually in the form of a bounding box containing the object. Images might contain object instances of the same classes, different classes, or no instances at all. And the detector must ensure that false detection is never made when no instance is present. The instances may vary in terms of size, lighting, rotation, appearance, *etc.*

The widespread deep learning based frameworks for object detection can be divided into two main categories:

1. Two-stage detectors (region proposal based approaches), basically include R-CNN [61], SPP-Net [62], Fast R-CNN [63], Faster R-CNN [3], R-FCN [64], FPN [65] and Mask R-CNN [66]. These methods follow traditional object detection pipeline, by decomposing the object detection problem into two main stages: (i) generate region proposals; (ii) classify each proposed region, and regress its bounding box coordinates.

2. One-stage detectors, basically include YOLO [67], SSD [68], YOLOv2 [69], DSSD [70], DSOD [71] and Libra R-CNN [72]. In these detectors the tasks of object localization and classification are done in a single forward pass of the network. This group of object detectors resorts to unified frameworks to generate final object detection results directly from image pixels. These methods do not rely on region proposals. Instead, a set of default boxes over different aspect ratios and scales is used and applied to the feature maps, which can shrink time cost.

Figure 2.5 illustrates pioneer object detection frameworks, which will be reviewed in next sections. In the following, we list the most important region proposal approaches and object detection frameworks.

2.3.1 Region proposal

Region proposal methods were first implemented as external modules separated from the detectors to generate candidate regions or bounding boxes. The main approaches are divided into three categories:

- Sliding window approach is the most straightforward way. In this approach, many possible windows in the image are successively considered and classified to decide if the window contains an object or not. This type of approach is successful but extremely costly. *Objectness* [73, 74] is one of the first sliding window methods for generating object proposals, it is considered as a low-level pre-processing stage, to propose a small number of windows likely to cover all objects in the image. The idea is to sample and rank a considerable number of windows per image according to their likelihood of containing an object based on different factors such as the saliency, edges, superpixels, color and location.

- Grouping super-pixels is the second type of approaches to generate candidate regions. This type of methods typically depends on an initial over-segmentation. Then different merging strategies are appointed to group similar segments into object proposals. *Selective*

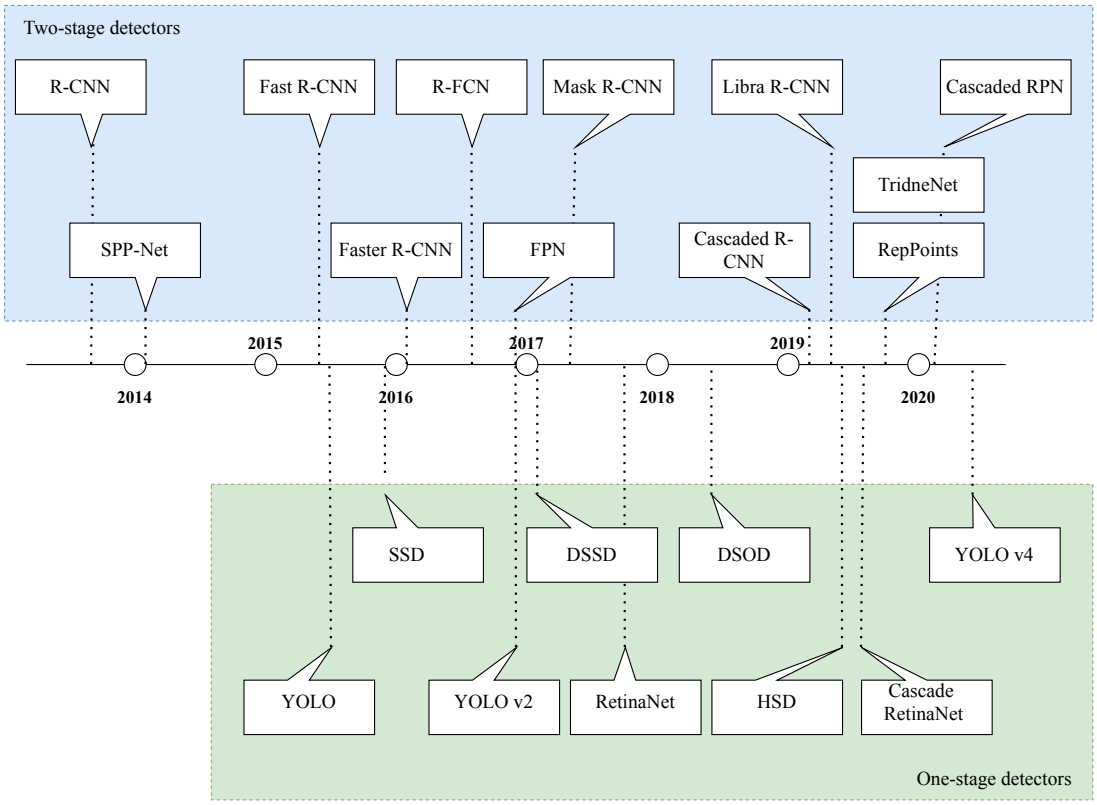


Figure 2.5 – Timeline of leading object detection frameworks into their two categories: (1) two-stage and (2) one-stage.

Search [75] is one of the most popular grouping super-pixels methods, developed as an alternative to comprehensive search in an image to localize objects. It is a clustering-based approach, which attempts to group pixels in regions based on engineered low-level features. The output is a few region proposals which may contain an object.

- Learning-based approaches, where the features are obtained using complex learning methods like deep learning models. *MultiBox* [76] is a CNN-based region proposal solution for detection, which predicts a set of boundary boxes and confidence scores for each of them. It can effectively match the performance of hand-engineered methods while allowing an efficient trade-off between runtime and quality. *Region Proposal Networks (RPN)* [3] inspired by MultiBox, predicts rectangular object proposals. Each proposal is tied up with an objectness score, based on a set of pre-defined reference boxes (anchors), each anchor is associated with a scale and aspect ratio. To recover the final bounding box, the regressed offsets must be added to the anchor. RPN shares convolutional layers with the object detection network, leading to a small marginal cost for computing the proposals.

2.3.2 Two-stage detectors

- **Region-Convolutional Neural Networks (R-CNN)** [61] By adopting selective search [75] R-CNN generates about 2000 region proposals for each image. Then each of these region proposals is scaled to a fixed size and fed into a CNN to extract its features. Finally, the output features of each CNN is classified with linear SVMs. Even though R-CNN significantly improved object detection performance, it was considerably time-consuming given the dispensable feature extraction for overlapped regions. Thereafter SPP-Net solved this problem.

- **Spatial Pyramid Pooling Network (SPP-Net)** [62] Unlike convolution layers, fully connected layers take as input a fixed-length vector. Derived from this idea, SPP-Net generates a fixed dimensional mid-level representation, which helps to use multi-scale images

efficiently. In SSP-Net a spatial pyramid pooling is performed on the last convolution layer on regions of arbitrary sizes, to generate feature vectors of constant size, irrespective of the input image size. These vectors are then fed to the fully connected layer for training the detectors. This method avoids computing the convolutional feature maps several times, and reuses them for efficient region-based object detection. This results in SPP-Net achieving the same performance as R-CNN 20 times faster. Although its speed, its drawback is: feature extraction network was not trainable.

- **Fast R-CNN** [63] Fast R-CNN replaced SPP layer by Region of Interest (RoI) pooling layer, and upgraded R-CNN detection speed by (i) replacing the different classifiers with a softmax layer; (ii) performing feature extraction over the image, then generating region proposals based on the last feature map of the network, not from the original image itself. Thus, training only one CNN over the entire image. Features of multiple bounding boxes within the same image are warped from the same feature map efficiently via RoI pooling operations. Even though Fast R-CNN effectively enhanced both R-CNN and SPP-Net, its detection speed is still restricted by generating the proposals.

- **Faster R-CNN** [3] It is the first end-to-end CNN-based object detection framework. The principle is to use Fast R-CNN feature maps in the Region Proposal Network (RPN), making it possible to propose regions using CNNs at no cost, eliminating the need for the slow region proposal algorithms. This is done by sharing convolutional layers in both region proposal and classification.

- **Region-Fully Convolutional Network (R-FCN)** [64] To adapt to the fully convolutional state of the art architectures, it's natural to construct a fully convolutional object detection network without a RoI-wise subnetwork. In R-FCN, the last convolutional layer produces position-sensitive score maps. Then a position-sensitive RoI pooling layer combines the responses from these score maps. Finally, in each RoI, the position-sensitive scores are averaged to produce a vector, and softmax responses across categories are computed.

- **Feature Pyramid Network (FPN)** [65] It is composed of a top-down pathway, a bottom-up pathway, and several lateral connections. The bottom-up pathway is the usual forward backbone CNN for feature extraction; it produces a feature hierarchy by downsampling the corresponding feature maps. As we go up, the spatial resolution decreases. With more high-level structures detected, the semantic value for each layer increases. The top-down pathway constructs higher resolution layers from a semantic rich layer. While the reconstructed layers are semantic strong, the locations of objects are not precise after all the downsampling and upsampling. The lateral connections between reconstructed layers and the corresponding feature maps to combine low-resolution and semantically strong features with high-resolution and semantically weak features.

- **Mask R-CNN** [66] A framework for object instance segmentation. It detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. It extends Faster R-CNN by adding a branch for predicting an object segmentation masks in a pixel-to-pixel manner, in parallel with existing branches for classification and bounding box recognition. Mask R-CNN is simple to train and adds only small expenses to Faster R-CNN. However, it requires pixel-wise annotations.

- **Libra R-CNN** [72] This framework integrates three novel components to balance the training process: (1) IoU-balanced sampling, that extracts hard samples based on their Intersection over Union (IoU) with assigned ground-truth. (2) Balanced feature pyramid, which enhances the multi-level features using the same deeply integrated balanced semantic features. (3) Balanced L1 loss, which promotes crucial gradients, rebalances the involved classification, global and precise localization.

Many other methods have been proposed over the last three years. They have made significant progress in addressing different aspects. For example, Cascaded R-CNN [77] extended Faster R-CNN to a multi-stage detector through cascade architecture. Cascade Region Proposal Network (Cascade RPN) [78] is aimed at improving the region-proposal quality and the detection

performance by leveraging a single anchor per location and refining it in several steps. Relation networks for object detection [79] is focused on exploiting the relationships between object instances during learning, rather than recognizing each of them independently. Representative points (RepPoints) [80] proposed non-rectangular representations for anchor-free object detection, which produce fine-grained localization. Trident Network (TridentNet) [81] addresses scale variation by generating scale-specific feature maps. Using several branches share parameters and have the same network structure, but have different receptive fields.

2.3.3 One-stage detectors

- **You Only Look Once (YOLO)** [67, 69] As a direct development of MultiBox, YOLO was the first CNN-based one stage object detector. In this approach, object detection is framed as a regression problem to spatially separate bounding boxes from image pixels, and associate class probabilities to each of them. A single CNN simultaneously predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. YOLO quickly learns general representations of objects, but it makes more localization errors than two-stage detectors. However, the latest version YOLOv4 [82] has proven to be more efficient, offering the best accuracy and speed; it has confirmed the viability of one-stage anchor-based detectors.

- **Single Shot MultiBox Detector (SSD)** [68] Unlike MultiBox, every feature map location in SSD is associated with a set of default bounding boxes of different aspect ratios and scales. These priors are accurately chosen by hand, whereas in MultiBox, they were chosen because their IoU with respect to the ground truth was over 0.5. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes.

- **Deconvolutional SSD (DSSD)** [70] By adding additional deconvolutional layers to SSD, DSSD provides a way to build feature pyramids to achieve multi-scale representation, includes high-level context, and improves accuracy, particularly for small objects. As an alternative of the element-wise sum, DSSD adds each layer, through element-wise products to its previous layer.

- **RetinaNet** [83] According to Lin *et al.* the great imbalance between foreground and background regions is why two-stage detectors always give better performance than the one-stage detectors. This makes them replace the cross-entropy loss by focal loss which emphasis on the hard misclassified examples during training.

- **Deeply Supervised Object Detectors (DSOD)** [71] This approach succeeded in learning object detectors from scratch, which draws attention to the importance of network design to get rid of the requirements for perfect off-the-shelf pre-trained classifiers on relevant tasks. The key discovery in DSOD is that deep supervision, enabled by dense layer-wise connections, plays a key role in learning a good detector. DSOD provides an encouraging direction to train from scratch to crossover the gap between classification and detection tasks.

Recently, new approaches have been proposed to improve one-stage detectors. For example, [84] designed two novel loss functions for balancing the gradient flow for anchor classification and bounding box refinement. Hierarchical shot detector [85] proposed to improve the accuracy by performing regression and classification hierarchically instead of simultaneously. [86] proposed a feature enrichment scheme to produce multi-scale contextual features to address the class imbalance problem and improve classification and regression. [87] showed that the misalignment of the optimization between training and inference is the bottleneck to achieve better results. Therefore, they proposed a consistent optimization to increase performance. There are many other methods and approaches such as RefineDet [88], cascade retinanet [89], and fully convolutional one-stage (fcos) object detector [90].

Anchor-free approaches. Anchor boxes and bounding boxes regression need a large number of anchors to ensure a sufficiently high Intersection over Union (IoU) rate with the ground-truth objects, The hyperparameters, such as scales and aspect ratios, are usually heuristically tuned and have a large impact on the final accuracy). To avoid these disadvantages, anchor-free approaches are proposed. For example, keypoint-based object detection approaches like CenterNet [91] inspired by CornerNet [92], detects each object as a triplet of keypoints *i.e.* center, top-left and bottom-right corners, which improves both precision and recall. [93] compare anchor-based and anchor-free methods, and provide a new method to automatically select positive and negative samples according to the object characteristics.

2.3.4 Application in the chairlift safety problem

The chairlift safety problem could be formulated as an object detection problem by detecting safety-relevant objects, in particular objects that are invariant to a change of chairlift. Instance-level bounding box annotations are required with sufficient variability to ensure that the trained model is robust to changing conditions and chairlift variations. Since these annotations are expensive, we anticipate training a cross-domain model, when trained on some chairlifts, could perform well in new unseen ones. That brings us to *domain adaptation* that will be discussed in the following section.

2.4 Domain adaptation

In the chairlift safety problem, models trained on some chairlifts may need to be adjusted to perform well in new unseen ones. In machine learning, generally, the test data is drawn from the same distribution as the training data, and have identical feature representation *i.e.* training and test sets belong to the same domain. A standard machine learning model trained on a *source* domain will perform poorly on a *target* domain, if the two domains are different. In this case, we refer to domain adaptation (DA), a particular case of transductive transfer learning (TL), where the task is the same (output space) in the source and target domains; however, the feature space or the data distributions are different but related to some extent [94]. Such differences referred to as domain shift, are known to occur in real-world applications because of many factors *i.e.* in visual applications: variation in background or location, viewpoint, lighting condition, image quality, *etc*. According to different domain shifts *i.e.* data distributions or feature space, DA can be divided into two main types *i.e.* homogeneous and heterogeneous (respectively) [95]. The key challenge is to bridge the gap between the domains, to the extent that a system trained on the source domain will generalize well to the target domain. Typically, in the source domain the data is labeled, where in the target domain there is either a small set of labeled examples (insufficient to train a model in the target domain) and here we refer to semi-supervised DA, or no labeled data is available at all and in this case, we refer to unsupervised DA [96].

In the chairlift safety problem, each chairlift is considered a domain, and models trained on one domain need to be adapted to handle a new unseen one successfully. This requires either annotating the data in the target domain *i.e.* the new chairlift, or adapting the pre-trained models to achieve suitable performance in the new chairlift. Labeling data is costly, time-consuming, and involves a significant human effort. However, adapting pre-trained models is a potentially smarter solution, which could be realized by exploiting previously acquired labeled source data and new unlabeled target data together. In this case, the domain divergence relates only to the distribution of the data, where the feature spaces are identical, thus, the required DA is homogeneous DA. Therefore, in the next section we will introduce homogeneous DA approaches, but first, it is pertinent to introduce Generative Adversarial Networks, as it is embedded in many of the deep domain adaptation methods that we will review later.

Generative Adversarial Networks

Generative adversarial networks (GANs) [97] proposed by Goodfellow *et al.* set two well-matched networks against each other *i.e.* a generator and a discriminator, from which comes the term 'adversarial' (See Figure 2.6). GANs were originally used to generate synthetic images, but other innovative uses have recently been investigated such as domain adaptation.

The generator G is trained to generate a vector that cannot be distinguished from the actual training data in a manner that confuses the discriminator D. In his turn, D is trained to provide a binary label that discriminates between the samples from G and the samples from the training data. The problem is formalized as mini-max optimization problem *i.e.* training G to minimize the loss (attempting to fool D) while also training D to maximize the loss (D tries not to be fooled). In DA the principle is to align distributions in the source and target domains, to have the network unable to distinguish between domains.

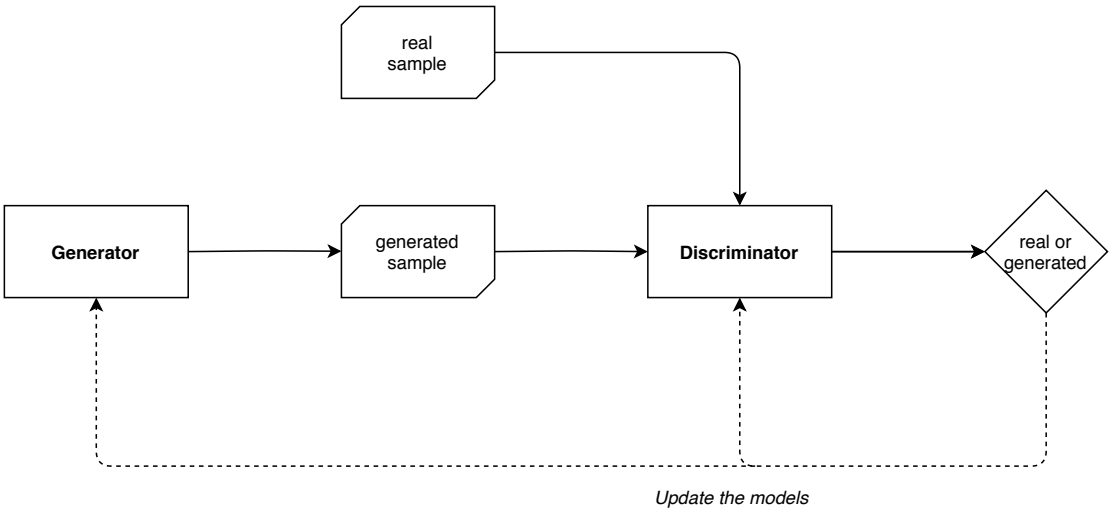


Figure 2.6 – Generative adversarial networks principle.

2.4.1 Homogeneous domain adaptation approaches

In this thesis, we focus our attention on deep learning; therefore, we will consider in this section deep DA approaches *i.e.* that employs a deep network to boost DA performance. The integration of the robust hierarchical representations of deep learning with domain adaptation allows the model to learn more transferable representations mapping both domains to mitigate the adverse effects of the domain shift. According to the survey conducted by Wang and Deng [95] about deep domain adaptation methods for computer vision applications, homogeneous domain adaptation approaches could be divided into the following three main categories:

1. **Discrepancy-based models** involve techniques for reducing the difference between domains, using statistical techniques on the corresponding activation layers of source and target networks (*e.g.* maximum mean discrepancy (MMD), correlation alignment (CORAL), batch normalization, *etc.*).

   Fine-tuning can be viewed as a discrepancy-based deep DA approach. It consists essentially of training a network on the source domain, then cutting off the 'head' of the network (usually the final set of fully connected layers) and replacing them with randomly initialized layers, the resulted network is then trained on the target domain. During training, the pre-trained layers (usually the convolutional part) of the network could be either frozen (their weights cannot be updated), or fine-tuned according to the target dataset size and its similarity to the source dataset. The flexibility and generality of deep learning-based models are commendable, as a result, fine-tuning has become a popular strategy for training deep learning-based models. Nevertheless, it is still difficult and expensive to obtain sufficient labeled data in the target domain to successfully fine-tune the deep networks (even when the convolutional part is frozen, the fully connected layers have a large number of parameters).

2. **Adversarial-based models** involve techniques for reducing the difference between the source and target domains, using an adversarial objective with respect to a domain discriminator. Two major categories of adversarial-based models can be identified:

- Generative models that generate annotated synthetic target images from the source images and train the network on these synthetic target data.

  Liu and Tuzel proposed coupled generative adversarial network (CoGAN) [98]. As the name suggests, CoGAN consists of two GANs: GAN_1 for source data and GAN_2 for target data. Source and target feature representations are jointly learned via weight sharing in the first few layers of the generative models and the last few layers of the discriminative models. By generating synthetic target data and synthetic source data, that share the labels, CoGAN achieves a domain-invariant feature space without supervision in the target domain.

  Long *et al.* proposed conditional adversarial domain adaptation (CGAN) [99]. Also based on GAN, CGAN extends to a conditional model, where the discriminative model is conditioned on the cross-covariance of domain-specific data representations and class labels.

- Non-generative models that try to match the feature distributions in the source and target domains either by finding a transformation between the domains or by directly adapting the features without generators.

  Ganin and Lempitsky introduced a domain-adversarial neural network (DANN) [100] with Gradient Reversal Layer (GRL) that attempts to match source and target feature distributions by jointly optimize the class predictor and the source-target domain disparity by back-propagation. GRL treats domain invariance as a binary classification problem, and directly maximizes the loss of the domain classifier by reversing its gradients.

  Tzeng *et al.* proposed adversarial discriminative domain adaptation (ADDA) [101] framework. First, the target model parameters are initialized by the pre-trained source model weights. Then a discriminative mapping of target domain images to the source domain feature space is learned by a domain adversarial loss. This enables learning more domain-specific feature extraction.

3. **Reconstruction-based models** involve techniques for reducing the difference between the source and target domains by creating a shared representation between the two domains while keeping the individual characteristics of each domain. (*e.g.* encoder-decoder models, dictionary and sparse coding models, and graph-based models).

   Zhu *et al.* introduced Cycle GAN [102] based on GANs, mapping inputs data from the source domain to target using cycle-consistency constraint to learn the relations between the domains in the absence of any paired training input-output instances. Cycle GAN employs two generators to learn mapping and inverse mapping. And two discriminators determine the quality of the realism of the image generated by an adversarial loss, and the quality of the reconstruction of the original input after a sequence of two generations by a cycle consistency loss.

2.4.2 Domain adaptation for object detection

Domain adaptation methods in the literature are traditionally applied to image classification tasks, and relatively few works address domain adaptation related to other computer vision tasks such as semantic segmentation, object detection, pose estimation, *etc.*, especially in the unsupervised setting. Probably a major reason for this is that these problems are more complex and often entail additional requirements (*e.g.* localization accuracy required for detection, pixel-level accuracy in case of image segmentation, *etc.*).

In the context of object detection, CNN-based models are typically initialized by deep pre-trained models with image-level annotations (often on ILSVRC datasets). When instance-level labels are available, the model can be further fine-tuned using the bounding boxes labels to improve both object identification and localization. This is known as supervised domain adaptive object detection, and it is the most straightforward case. Below we review the advanced approaches in the literature addressing the more challenging problem when little or no labelled target data is available.

Weakly supervised domain adaptation for object detection [2]. In this case, the source domain is well annotated with bounding box annotations, while the target domain is either partially annotated with bounding box annotations (semi-supervised) or annotated with image-level annotations.

One of the earliest works on domain adaptation for object detection is large-scale detection through adaptation (LSDA) [103] proposed by Hoffman *et al.* LSDA presented how to transform an image classifier trained on large-scale image datasets into an object detector requiring bounding boxes annotations for only a subset of categories in training. This is achieved by transforming the task into a domain adaptation problem, considering the data used to train the classifiers (images with category labels) as the source domain and the data used to train the detectors (images with bounding boxes and category labels) as the target domain. Using a CNN pre-trained on image classification, they replace the classification output layer with K linear classifiers, one for each category of the detection task. Then they fine-tune the first layers for the target domain using the available labeled detection data for B categories ($B \ll K$). Finally, they learn a category-specific transformation by finding for A categories the nearest neighbor category among B ($A = K - B$). The advantage of this model is that it generalizes well even for the classes for which there were no bounding box annotations during the training phase.

Raj *et al.* employs subspace alignment using Principal Component Analysis (PCA), to adapt R-CNN to adjust class-specific representations of the bounding boxes between the source and target domain [104]. The source bounding boxes are extracted from the well-annotated training set, while the target bounding boxes are obtained with the R-CNN detector trained on the source set. The detector is then re-trained with the target aligned source features and used to classify the target data projected into the target subspace.

Inoue *et al.* tackle the problem of cross-domain weakly supervised object detection using a progressive technique [105]. At first step, a detector is pre-trained in the source domain using instance-level annotations. At the second step, the detector is fine-tuned with artificial samples generated by domain transfer (transferring images with instance-level annotations from the source domain to the target domain), and the prediction probability of this step is used as a measure of confidence, only high-confidence detections are used for the next step. At the third step, the detector is fine-tuned with pseudo-labeling samples (pseudo-labeling the images with image-level annotations in the target domain).

Unsupervised domain adaptation for object detection. In this case, annotations are available only in the source domain, where only unlabeled images are available in the target domain.

The use of synthetic data which is available (almost) for free, to enrich the real data has become popular since the massive adoption of deep CNNs to address computer vision problems that require large amounts of annotated data. DA methods align a model trained with the synthetic (source) data to the real (target) data, especially when no or few labeled samples are available in the real domain. For example, Chen *et al.* proposed Domain Adaptive Faster R-CNN (DA Faster R-CNN) [106], which uses adversarial training to adjust features between real and synthesis domains at two different levels of Faster R-CNN architecture. The adaptation at the image level intends to eliminate the domain distribution discrepancy at the output of the backbone network, while the instance level adaptation concerns the features which are pooled from the RoI layer, before the final category classifiers.

Yu *et al.* proposed DALocNet [107] to improve the localization accuracy of DA Faster R-CNN. Using the highest confidence score pseudo labels produced by Faster R-CNN in the target domain and a weighted loss function to train the network on the target domain and learn it to better localize the objects. Also, inserting residual blocks into the shallow layers of a CNN is used to extract features in the target domain to enhance discriminative feature extraction, which helps for object localization.

Saito *et al.* argue that a global matching may adversely affect the performance for large domain shifts because domains can have different scene layouts and different combinations of objects, while local matching does not change category level semantics [108]. They proposed to combine a strong alignment of local features and a weak alignment of global ones, following the same adversarial training approach.

[2] Weak supervision *i.e.* incomplete, inexact, inaccurate or noisy supervision

Shan *et al*. proposed an unsupervised domain adaptation method to minimize domain shift at both the pixel and feature level [109]. Adaptation in image pixel space is achieved by using GAN objectives. The other adaptation component is added to Faster R-CNN for adversarial domain training at the last layer in the shared convolutional block; therefore, the features could be invariant in the domain, but the proposals are not adapted, resulting in limited localization capability.

Zhu *et al*. proposed a domain adaption model for object detection focusing on region mining *i.e*. clustering strategy to identify the most critical local regions for the source and target domains, in order to improve efficiency [110]. And then they apply the alignment between the two domains at the regional level in an adversarial manner.

Hsu *et al*. proposed to narrow the significant domain gap between source and target by using an intermediate domain instead of direct mapping, *i.e*. break down the adaptation into two steps that both solve a more manageable problem with a smaller domain gap, to give more attention to individual discrepancies in each step [111]. First, the source images are transformed to match the appearance of the target using the CycleGAN image-to-image translation network. Next, adversarial learning is employed to align the resulting synthetic images with the target domain at the feature level.

We note that all the works mentioned in this section used Faster R-CNN detector as a baseline, and they tried to add their adaptation components in different manners at different levels, mainly using adversarial training. But none of them tried to adapt the Region Proposal Network (RPN), leading to a residual domain shift; we propose to solve it in Chapter 4.

2.4.3 Beyond domain adaptation

Several methods closely related to domain adaptation address several domains or transfer the acquired knowledge to tasks for which few or no examples are given. Some of these methods are discussed below.

Domain generalization

Domain generalization is a transfer learning problem, referring to learning representations that integrate knowledge from several related source domains into a model for a new previously unseen target domain [94]. Unlike DA where unlabeled examples from the target domain are available, in domain generalization, no examples from the target domain are provided to adapt the model. The representations learned in domain generalization are efficient, but they do not consider any additional information that domain features may imply about the labels.

Muandet *et al*. proposed domain-invariant component analysis (DICA) [112], a kernel-based optimization algorithm that learns a mapping that minimizes domain dissimilarity, thus enhancing the expected generalization ability of classifiers on new domains.

Zhou *et al*. proposed to solve the problem of domain generalization by mapping source learning data to synthesized data from unseen domains [113]. This is accomplished with a learning objective formulated to minimize label classification loss while maximizing domain classification loss.

Shankar *et al*. proposed to learn a domain invariant classifier [114] using multi-domain training data, to generalize to unseen domains. Their original solution consists in artificially transferring data between domains to learn invariant features and they show that this augmented data helps to better generalize to unseen domains.

Few-shot learning

Few-shot learning is a particular case of transfer learning that occurs when only a few labeled examples of the target (domain or task) are available. The ultimate form is zero-shot learning (or zero-data learning) [115] when absolutely no labeled examples are given at all. In the same boat, only one labeled example is given for one-shot learning [116]. Few-shot learning is possible because additional information is involved during training; this additional information is usually semantic information about the unseen target domain or task.

Yang and Hospedales proposed associating each domain (source and target) with a vector of discrete parameters called a semantic descriptor [117]. Then, they use a two-branches network

whose inputs are the sample features and the domain descriptor from this sample. The output of the network (the predicted sample class) is a fusion of the outputs of the two branches. This is a way to adapt the classifier to the domain provided as input. For a new unseen target domain, given its semantic descriptor, the network is able to accurately predict the class of its samples. This approach requires that the user can describe all the domains with a vector of discrete parameters.

Kumagai and Iwata used a similar architecture, but instead of using an attribute vector for each domain, they proposed to extract a latent domain vector from the set of features of each domain [118]. This approach requires the knowledge of the whole target features to start predicting the classes of the target data.

Elhoseiny *et al.* proposed a model that can link textual information of visual categories to images with part based-regularization [119]. Using dimensionality reduction transformation from pure text description to part classifier, the classifiers are then applied to the visual part learning representation.

Multi task learning

Multi-task learning [120] tries to improve generalization by learning multiple tasks simultaneously with some shared internal representation, where source tasks are supervised, and target tasks are unsupervised. Yang and Hospedales method [117] mentioned before to solve a zero-shot learning problem could also be considered a multi-task problem.

Liu *et al.* proposed a multi-task deep neural network (MT-DNN) [121] to enhance the learning of text representations to improve the performance of several natural language understanding tasks. They state that combining a pre-trained language model with multi-task learning has a regularizing effect that leads to more general representations that improve adaptation to new tasks and domains.

Lee *et al.* attempted to predict some other information, in addition to the main classical detection task (prediction of the location and class of the objects), such as the area portions occupied by each ground truth box within a window, the distances from the center of the box to those of other boxes or a binary mask between foreground and background [122]. All these data were available from the ground truth labels but trying to predict them helped to solve the main detection task.

Likewise, Channupati *et al.* improved the results of their semantic segmentation network by adding a branch that estimates the depth of the pixels as an auxiliary task [123]. Since the depth was available in their used dataset, they proposed to exploit it at training time and create a multi-task network. At test time, they just removed the depth estimation branch and noticed that the main task (semantic segmentation) was improved. These last solutions are specific to the considered tasks and available data at training time. They can not be applied to our problem.

2.5 Guided problems under constraints

In the chairlift safety problem, we have at our disposal a knowledge of both the particular geometry of each chairlift, and the common information shared by all chairlifts. One direction for developing better deep learning-based systems is to make better use of the data itself, and there are many ways to leverage data and learn good representations. For example, using prior knowledge about the problem in terms of additional data or constraints will lead to better generalization, assuming that the use of this prior knowledge is justified. It is often argued that the recently demonstrated success of CNN-based approaches results from enhanced visual representation and large-scale use of data for learning. Therefore, in this section, we will review the approaches that try to explore how to better use the data to improve performance.

2.5.1 Visual attention

Attention is the intellectual process of selectively focusing on one or more things while ignoring others. One of the most remarkable aspects of human visual perception is the *attention*. Rather than processing a whole image in its entirety at once, attention allows to locate regions of interest and analyze the scene by selectively processing subsets of the image. This is particularly essential

when there is a lot of clutter in the image, to narrow down the search and speed up the process. Likewise, in neural networks the attention mechanism equips a neural network with the ability to focus on a subset of its inputs or features. Attention can be applied to any kind of inputs, regardless of their shape. In the case of matrix-valued inputs, such as images, we can talk about *visual attention*.

Attention has been applied in Deep Learning (especially, recurrent neural networks that include an encoder-decoder), to a wide variety of tasks *e.g.* machine translation [124, 125], speech recognition [126], image captioning [127], visual question answering [128], *etc.* RPN component in Faster R-CNN object detection framework is inspired by attention mechanism *i.e.* RPN tells the network where are the regions of interest to focus on. Below we review some work in the computer vision field, where visual attention mechanism has achieved great success, using *mask-guided* approaches.

Object detection

Zhao *et al.* introduced a model to enhance object detection with weakly-supervised object segmentation by sharing the same underlying convolutional features between the two tasks [129]. Their objective is to leverage object segmentation to improve object detection without relying on expensive pixel-wise ground truth segmentation masks. The segmentation information used is instance-level and only supervised by bounding box annotation. The segmentation ground truth binary object masks are unknown. First they are initialized from the object bounding boxes, and then recursively refined. The estimated object masks are called pseudo ground truth masks. The segmentation sub-network generates object masks, which are used together with the ground truth bounding box annotations to refine pseudo masks with a graph-cut refinement. In this approach, the generated object masks enhance the object detector with segmentation feedback.

Object segmentation

Dai *et al.* proposed to leverage shape information at a late stage in the network, by masking the last convolutional feature maps with a fine, irregularly shaped segment, instead of masking the raw image [130]. These segments given by the region proposal methods are considered as masks, and the resulted masked-feature maps are used then for training the classifier. The quality of convolutional features dramatically impacts the performance, therefore, computing them from the unmasked image help to preserve information that could be useful, which might be lost when masking the raw image.

Person re-identification and pedestrian detection

Song *et al.* introduced a binary mask guided contrastive attention model for the problem of person re-identification [131]. They use a binary body mask to eliminate background clutter at the pixel level. This can significantly improve the robustness of re-identification models under a range of background conditions. The mask contains body shape information and has been shown to be resistant to light, tissue colors and is therefore useful in identifying a person. The binary masks are used in addition to the input image; therefore, the model can learn the shape information from the masks and the other features directly from the image.

Pang *et al.* proposed a mask-guided attention network that concentrates on visible pedestrian regions while eliminating the occluded ones by modifying full body features [132]. The network comprises two parts: (i) standard pedestrian detection part providing features using full body annotations; (ii) mask-guided attention part generating a pixel-wise attention map using visible-region information, thus emphasizing the visible body region while eliminating the occluded part of the full body.

Co-saliency Detection

Zhang *et al.* introduced a mask-guided fully convolutional network (FCN) structure to initially generate co-saliency [3] detection result, which is further refined by a multi-scale label smoothing model [134]. The masks are added at different convolutional layers for background removal. For each image the mask is learned from the high-level feature maps in an unsupervised manner.

[3] "Co-saliency indicates the common and salient visual stimulus residing in a given image group." [133]

The learning objective maximizes the mask variance as well as promotes sparse mask inputs. Then, the learned masks are used to mask the convolutional feature maps of the FCN for the extraction of salient objects. Each feature map is masked with the learned mask by element-wise multiplication. Finally, the masked feature map is refined by optimizing a multi-scale label smoothing model. In this work, the experiments proved that the masks encode valuable semantic and spatial information and, therefore, the learned convolutional features are more discriminative with such guidance.

2.5.2 Conditional networks

Neural networks can be conditioned by additional information that expresses prior knowledge about the domain or the task. This additional information is combined in a hidden representation together with the input to define conditional distributions on functions. Incorporating generic constraints is an effective method for exploiting available data and regularizing learning, particularly in the presence of insufficient training data.

For example, Zhou *et al.* developed a network to estimate the 3D human pose from non-calibrated 2D images [135], and proposed to introduce geometric constraints to train their network. The constraints are based on the relative size of the human bones such as: upper and lower arms have a fixed length ratio, left and right shoulder bones share the same length, *etc.*

Zhao and Snoek proposed to modulate the RGB features of a video with optical flow features in order to improve the action detection accuracy [136]. The proposed motion condition and motion modulation layers incorporate motion and modulate the contribution of the RGB features. Such conditional networks require the different features (optical flow and RGB) to be well spatially registered.

2.5.3 Siamese networks

Siamese networks are a type of artificial neural network designed as two identical twin networks joined at their final layer by a distance layer. As the name suggests, the two networks have the same architecture, and all weights and biases are tied together. Each of these twin networks takes an input vector, and based on it an output vector is produced. The distance layer is trained to predict whether two inputs are of the same category or not, by computing how similar or dissimilar the corresponding outputs of the twin networks are. Siamese networks learn by training on pairs of images and force the distance to be zero for images of the same category and more than a threshold for images of different categories. This architecture is used to perform binary classification by computing the Euclidean distances between a test sample and the train samples of the two classes. The class with the highest similarity is decided as the predicted class. Siamese networks are an effective way to extend the knowledge and learning capacity of neural networks to small datasets. These networks have found their ways in a broad set of problems such as facial identification, signature verification, *etc.*

Bertinetto *et al.* proposed a fully convolutional Siamese network [137] to model visual tracking of objects as a similarity learning problem. By comparing the target image patch with candidate patches in a search region, it is possible to track the object to the location with the highest similarity score. A notable advantage of this method is that it learns solid embedding in an offline phase when little or no online training is required. Thus, real-time tracking can be performed easily. Siamese networks are widely used in the context of object tracking [138, 139, 140].

En *et al.* propose three-stream network (TS-net) [141] to exploit the advantages of the Siamese architecture to extract features which are shared by two modalities, and the pseudo-Siamese architecture (where only the architecture is identical, but no weight is shared) to compare patches from different modalities.

In our context, Siamese networks can be used to compare image features with features extracted from binary masks representing shape priors of the important objects in the scene.

2.6 Conclusion

In this thesis, we tackle the problem of monitoring chairlifts as a transport means, aiming at accurate and reliable scene analysis, while reducing maintenance costs and decreasing system response time. The intelligent visual surveillance involves many different disciplines, such as image classification, object detection, pattern recognition, *etc*. Developments in these research areas have raised expectations, especially in the age of deep learning which has been improving significantly in the last few years.

However, very few works have attempted to use a deep learning-based system in chairlift security by computer vision. Moreover, the main limitations of standard deep learning methods in our context are: (i) they require massive sets of annotated image for training, which is not always available in our case, (ii) it is not easy to determine which elements of the scene are used for the prediction, and (iii) they can have low generalization properties if there is a shift between the distributions of source and target domains.

To address these issues, we will first approach the chairlift safety problem as an object detection task to localize the safety bar in images and classify them into open or closed. Moreover, we will consider Faster R-CNN object detection framework because it has proved its efficiency where many advanced works take it as groundwork and have built on it many context-dependent improvements, especially in domain adaptation.

Due to the variations between the different chairlifts, each one could be considered independent domain, but close to other domains. So, when we have a new chairlift, we could use models trained on other chairlifts; however, we need domain adaptation components. Therefore, we will study the problem from a domain adaptive point of view, in order to evaluate the ability of the object detector trained on some chairlifts to perform well on a new unseen chairlift.

In the chairlift safety problem, we have at our disposal a knowledge of both the particular geometry of each chairlift, and the common information shared by all chairlifts. One direction for developing a better system is to better use the geometrical constraints at our disposal by guiding a classifier using binary masks representing the safety bar.

In the later chapters, we investigate each of these aspects, present our empirical experiments and proposed methods to address the problem of chairlift safety, and our contributions in the research areas being studied.

Chapter 3

Performance and Limits of Object Detection in the Chairlift Safety Problem

Unlike most existing solutions that address the chairlift safety problem as a global image classification task, we propose instead to focus on specific local features in the scene, such as the features of the safety bar and the people. A straightforward approach is to apply a classical object detector such as Faster R-CNN [3] to detect those essential elements in the scene. The object detector is a solution to guide the network to the crucial regions of the image. Then, at test time, we can use it as a classification model by ignoring the location of the bounding boxes. In this chapter, we present the results provided by such a detector on our chairlifts dataset for the detection of the safety bar (open/closed), and the detection of people (adult/child).

3.1 Introduction

In the context of Mivao project, a low-level analysis should be performed for the videos monitoring the boarding station. This analysis aims to detect important elements of the scene, such as the people and the safety bar. Different image and video analysis algorithms could be relevant and useful when performing this kind of analysis. Since our dataset is composed of a sequence of images, we are more concerned about image analysis. In this chapter, we have decided to approach the problem as an object detection problem. Because when the object is small with respect to the entire image (like the safety bar), that will result in a weak signature in the global image representation. In that case, we think that better performance would be achieved with object detection (even if the exact location of the object is not important) because the features are aggregated in a more regional level of the image.

One of the most important state of the art object detection models is Faster R-CNN [3] (as explained in Section 2.3). Although it is not the fastest or the most efficient approach today, it remains an essential foundation for deep object detection models. Furthermore, the early researches in domain adaptation for object detection area were based on Faster R-CNN (*e.g.* [106] and [105]).

The first experiments carried out in this thesis, consist in applying Faster R-CNN on our chairlifts dataset to evaluate object detection in the chairlift safety problem, determine if it is a valid solution, and identify the weaknesses of this approach. The chapter is organized as follows. In Section 3.2 we overview Faster R-CNN approach, the experimental settings and the evaluation metrics. In Section 3.3 we present the implementation details of safety bar detection, results, and examples. Following the same scheme in Section 3.4, we present people detection results. We conclude and draw future directions in Section 3.5.

3.2 Faster R-CNN in the chairlift safety problem

Given that we want to detect dangerous situations concerning people boarding a chairlift, we can formulate the problem as object detection of the important elements of the scene *i.e.* the people and the safety bar. First, we will consider the safety bar detection (classes: open, closed), then people detection (classes: adult, child). Thus, as an output of the model, we intend to have a set of labeled bounding boxes representing each instance in the image, as illustrated in Figure 3.1. In all our experiments, we use the original python implementation of Faster R-CNN [1].

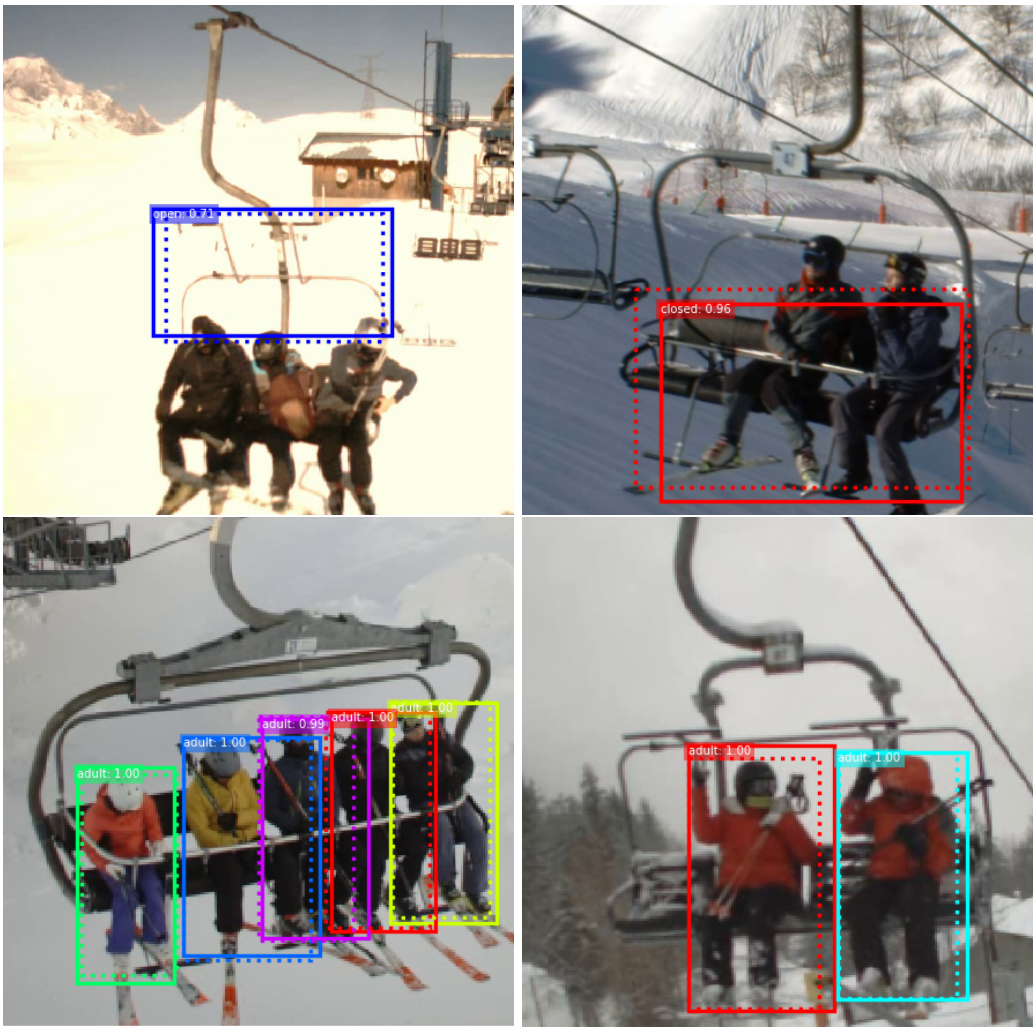
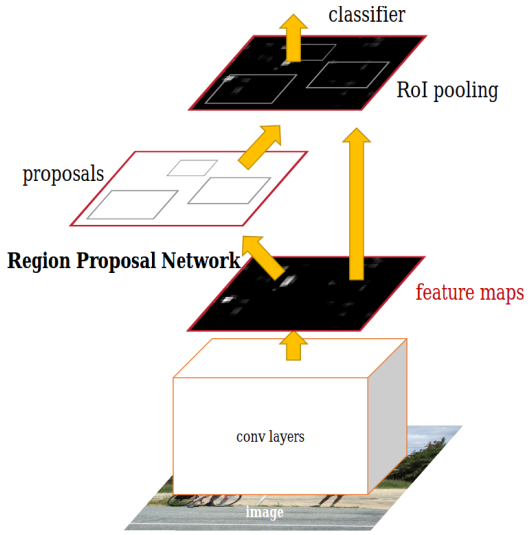


Figure 3.1 – Examples of the desired output. The dotted line represents the ground truth and the solid line represents the detection. Top: safety bar detection, Bottom: people detection.

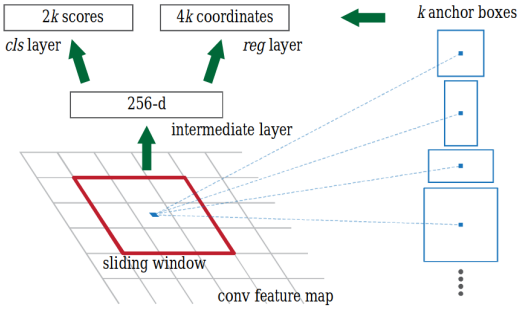
3.2.1 Faster R-CNN model

Faster R-CNN [3] is a foundation model for deep learning-based object detection. As its name suggests, it is faster than its ancestors R-CNN [61] and Fast R-CNN [63]. Faster R-CNN is a unified network for object detection composed of two modules: Region proposal network (RPN) and Fast R-CNN detector (see Figure 3.2a). RPN indicates to Faster R-CNN where it should focus its attention to detect objects. The main advantage of Faster R-CNN is that (RPN) and Fast R-CNN detector share the first block of convolutional layers, which makes the region proposal step at no additional cost. Below we will discuss in details the two modules, and the shared features.

[1] https://github.com/rbgirshick/py-faster-rcnn



(a) Faster R-CNN unified object detector.



(b) The mini-network at a single position

Figure 3.2 – Faster R-CNN model (Figures from [3]).

Fast R-CNN

Fast R-CNN model is made up of the following steps:

1. Processes the whole image with several convolutional and max pooling layers to produce a convolutional feature map.

2. For each object proposal (also called region of interest RoI), a RoI-pooling layer extracts a fixed-length feature vector from the feature map. We denote u the true class label for RoI, $u \in [0, K]$, $u = 0$ for background by convention, and v the true bounding box for class u .

3. Each RoI is fed into a sequence of fully connected (FC) layers that finally branch into two sibling output layers. One produces softmax probability over K object classes in addition to the 'background' class *i.e.* $p = (p_0, ..., p_K)$. And another layer outputs 4 real-valued numbers for each of the K object classes $t = (t^0, ..., t^K)$, encode refined bounding-box positions.

The final Fast R-CNN loss function for a RoI is a multi-task loss $\mathcal{L}$, defined as:

$$\mathcal{L} = \mathcal{L}_{cls}(p, u) + \lambda[u \geq 1]\mathcal{L}_{loc}(t^u, v) \tag{3.1}$$

Where, $\mathcal{L}_{cls}$ is log loss:

$$\mathcal{L}_{cls}(p, u) = -\log\, p_u \tag{3.2}$$

and $\mathcal{L}_{loc}$ is L_1 loss:

$$\mathcal{L}_{loc}(t^u, v) = \text{smooth}_{L_1}(t^u, v) \tag{3.3}$$

λ is a hyper-parameter to balance the two losses (set to 1 in all experiments), the term $[u \geq 1]$ is to ignore $\mathcal{L}_{loc}$ for background RoIs, because they do not have ground truth bounding box notion.

By neglecting the time of generating region proposals, Fast R-CNN achieved near real-time rates using very deep networks. Therefore, region proposals were the computational bottleneck at test time. The solution came through Faster R-CNN, which employs the same convolutional feature maps used by Fast R-CNN, for generating region proposals. According to that, the slow Selective Search algorithm was replaced by the fast deep fully convolutional RPN module, which takes the role of proposing regions. Then Fast R-CNN module uses the proposed regions to detect objects, and no more need for an external method for candidate region proposals.

Region proposal network (RPN)

RPN is a fully convolutional network predicts: (i) the position of the object t by ranking region boxes and proposing the ones most likely contain objects; and (ii) the "Objectness" score o at each position. Taking an image of any size as input RPN outputs a bunch of rectangular box proposals, with an objectness score for each image region as follows: a mini-network (illustrated in Figure 3.2b) moves across the last feature map of the shared convolutional network in a sliding window manner based on a set of pre-defined k reference boxes (anchors). Each anchor is assigned a scale and aspect ratio for efficiently detecting objects with a wide range of scales and aspect ratios. The mini-network maps each $n \times n$ spatial window of the feature map to a lower dimension vector and fed it into two siblings 1×1 fully-connected layers shared across all spatial locations, those two layers are:

- a box-regression layer (reg) recover the final bounding box coordinates of k anchors by adding the regressed offsets to the k anchor centered at the sliding window in question;
- a box-classification layer (cls) outputs $2k$ scores represent the softmax probability that estimates whether each of the k proposals contains an object.

For training RPNs, a binary class label (of being an object or not) is assigned to each anchor. The basic idea to label the anchors using the ground truth boxes, is to label the anchors having the highest Intersection-over-Union (IoU) overlap with ground truth boxes as foreground (positive label), and the ones with the lowest overlaps as background (negative label). Finally, the anchors that are neither positive nor negative do not take part in the training. The final RPN loss function for an image is a multi-task loss, defined as:

$$\mathcal{L} = \frac{1}{N_{cls}} \sum_i \mathcal{L}_{cls}(o_i, o_i^*) + \lambda \frac{1}{N_{reg}} \sum_i \mathcal{L}_{reg}(t_i, t_i^*) \tag{3.4}$$

Where, i is the index of an anchor in a mini-batch, o_i the predicted probability of anchor i being an object, o_i^* ground truth binary label of whether anchor i is an object or not, t_i predicted four coordinates; t_i^* ground truth coordinates, N_{cls} normalization term, set to the mini-batch size (256), N_{reg} normalization term, set to the number of anchor locations (2400).

Anchor conception presents a new scheme to handle multiple scales and aspect ratios. It relies only on single scale images and feature maps. It uses sliding windows of a single size on the feature map, avoiding adding images or filters of different scales or aspect ratios, which is more cost-effective.

Sharing Features and architectures

Faster R-CNN employs a fully convolutional network for feature extraction. The resulting feature maps are then used to generate region proposals. The same feature maps together with the proposed regions are used by the object detection network.

The convolutional layers of RPN and Fast R-CNN could be trained independently or jointly. In approximate joint training, the RPN and Fast R-CNN are considered as a unified network. During training, RPN generates region proposals, and the Fast R-CNN detector considers them as fixed and pre-calculated proposals (the derivative with respect to the bounding box coordinates is discarded). Then, Fast R-CNN loss and RPN loss are backpropagated throw the shared layers:

$$\mathcal{L} = \mathcal{L}_{Faster} + \lambda \mathcal{L}_{RPN} \tag{3.5}$$

Deep CNNs are an ideal choice to extract deep and expressive features for both RPN and the object detection network in Faster R-CNN. In our experiments, we investigate two architectures:

- Zeiler and Fergus model [59] (ZF) which has 5 convolutional layers and 3 fully-connected layers, its convolutional part has a total of 4.4 million parameters. ZF is a simple network and fast to train.
- Simonyan and Zisserman model [60] (VGG16) which has 13 convolutional layers and 3 fully-connected layers, its convolutional part has a total of 17.1 million parameters. By increasing the number of convolutional layers and using smaller filters, VGG was able to achieve better performance than its ancestors.

We only use the convolutional layers, initialized by pretraining the model on ImageNet [142] classification task as a standard practice. All remaining layers in Faster R-CNN are randomly initialized.

3.2.2 Experimental settings

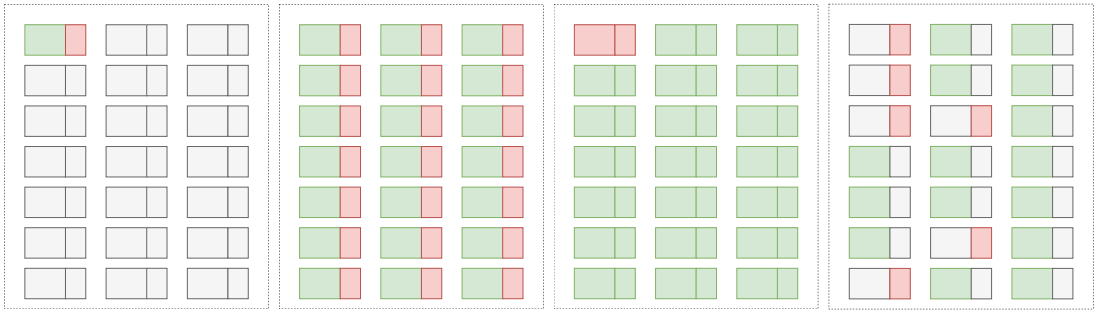
To study the behavior of the different architectures and compare them with the performance of our partners in the project, we considered four different experimental settings, of which Bascol *et al.* [34] proposed the first three (see Figure 3.3):

1. **OOC** ("Only One Chairlift"). The training and test sets are composed of images from a single chairlift. In this setting, we consider a dedicated model for each chairlift; the dataset is small with few variations.

2. **All**. The training and test sets are composed of images from all the chairlifts. Here, we have a common model for all chairlifts. Thus, the dataset is immense, with so many variations.

3. **LOCO** ("Leave One Chairlift Out"). The training set is composed of images from all the chairlifts but one, and the test set is composed of the images of the remaining chairlift. In this setting, we simulate the installation of a new chairlift to evaluate the performance of a model trained with all the previous chairlifts.

4. **FS** ("Fifteen-Six"). The training set is composed of images from 15 chairlifts training sets, and the test set is composed of the images of the remaining 6 chairlifts test sets. In this setting, we study the ability of the model to generalize to new unseen chairlifts (target domain) when it is trained on different chairlifts (source domain). This is basically the same objective as *LOCO* configuration, except that we have several chairlifts in the target domain where we can study the model's generalizability to a variety of chairlifts at the same time. We omitted the source domain test sets at training time so that we could also evaluate the model on the source chairlifts using these test sets. And at the same time, we kept a separate training subset for each target domain so that we could use these images later in domain adaptation.



(a) OOC (b) All (c) LOCO (d) FS

Figure 3.3 – Experimental settings. Each small rectangle represents one chairlift, the training set in green and the test set in red.

3.2.3 Evaluation metrics.

To validate the detection and assess the overall performance of a detection system, several metrics are used, which we indicate below.

- Intersection over Union (IoU) is a measure to evaluate the overlap between two bounding boxes, *i.e.* ground truth bounding box B_{gt} and predicted bounding box B_p (see Figure 3.4).

$$IoU = \frac{area(B_{gt} \cup B_p)}{area(B_{gt} \cap B_p)} \tag{3.6}$$

In the context of object detection, to consider the predicted bounding box as correct detection, the IoU between it and the ground truth bounding box should be higher than a threshold [2] see Table 3.1.

Table 3.1 – TP, FP, FN and TN in the context of object detection

$TP(c)$	True Positive	Correct detection IoU $\geq$ threshold	**a proposal** was made for class c, and there actually **was an object** of class c
$FP(c)$	False Positive	Wrong detection. IoU <threshold	**a proposal** was made for class c, when in fact there **was no object** of class c
$FN(c)$	False Negative	A ground truth not detected	**no proposal** was made for class c, when in fact there **was an object** of class c
$TN(c)$	True Negative	Does not apply	**no proposal** was made for class c, and there actually **was no object** of class c

- The precision is the ability of a model to identify only the relevant objects. It is the percentage of correct positive predictions among all the predictions. The precision for class c is given by:

$$Precision(c) = \frac{TP(c)}{TP(c) + FP(c)} = \frac{TP(c)}{all\ detections} \tag{3.7}$$

TP and FP are explained in Table 3.1.

- The recall is the ability of a model to find all the relevant cases (all ground truth bounding boxes). It is the percentage of correct positive predictions among all relevant ground truths. The recall for class c is given by:

$$Recall(c) = \frac{TP(c)}{TP(c) + FN(c)} = \frac{TP(c)}{all\ ground\ truths} \tag{3.8}$$

TP and FN are explained in Table 3.1.

- F-measure (F-score) is a harmonic mean of the precision and recall. F-measure, allows for different weights for precision and recall, but generally they are given equal weight, resulting in F_1. For class c, F_1 is given by:

$$F_1(c) = 2 \times \frac{Precision(c) \times Recall(c)}{Precision(c) + Recall(c)} \tag{3.9}$$

- Average Precision (AP) is the mean of the precision values at each confidence threshold n, weighted by the difference between the actual recall value and the previous one. AP for class c is given by:

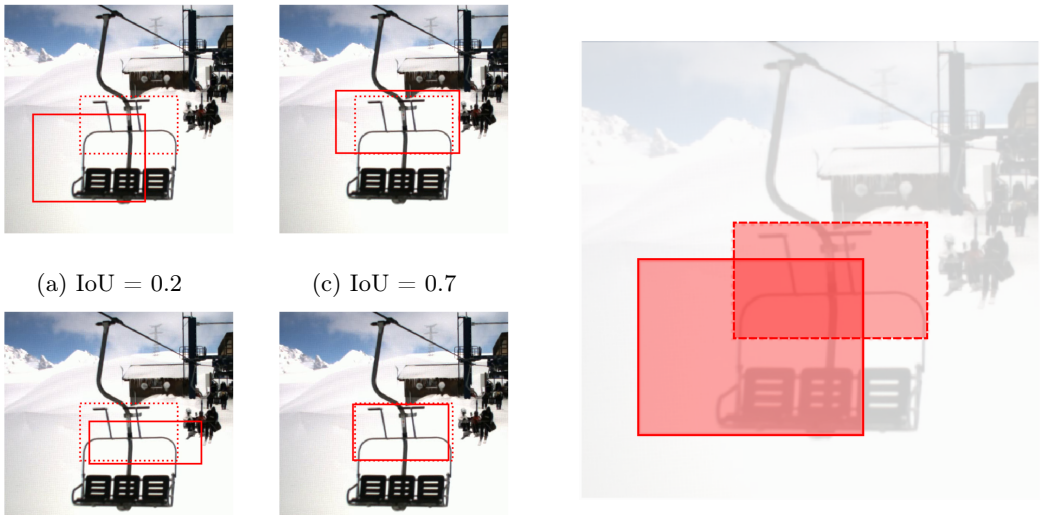
$$AP(c) = \sum_n (Recall_n(c) - Recall_{n-1}(c)) \times Precision_n(c) \tag{3.10}$$

AP is the area under the curve (AUC) of the Precision $\times$ Recall curve. In some cases, the precision is interpolated at a certain number of points *e.g.* 11, in this case 11-points interpolated precision is an approximation of AUC.

Mean Average Precision (mAP) is the AP averaged across all classes. mAP is given by:

$$mAP = \frac{1}{|classes|} \sum_{c \in classes} AP(c) \tag{3.11}$$

[2] Usually set to 50% or 70%.



(a) IoU = 0.2 (c) IoU = 0.7

(b) IoU = 0.5 (d) IoU = 0.9

IoU=*overlap*/*union*

Figure 3.4 – The overlap and union for the ground truth bounding box (dotted) and different predicted bounding boxes (solid) leading to different IoU values.

Non-maximum Suppression

A standard object detection pipeline generates hundreds of proposals and assigns foreground-background scores to them. Overlapping candidate regions have similar scores because they share most of their features. In order to have a high recall, the proposal generation method keeps all these proposals. However, passing all these proposals through the classification sub-network is time-consuming and increases false positives. That is why a post-processing technique to reduce the number of proposals is needed, which brought us to the Non-maximum Suppression (NMS) [143]. The algorithm simply sorts the proposals by their confidence scores. Then add the proposal with the highest score to the output list. Then check its IoU with the neighborhood proposals and remove those having an IoU higher than a threshold, because these less-confident proposals are likely to cover the same object. This process is repeated on the remaining proposals until all of them are processed. When we set the NMS threshold to a value close to 1, we get a lot of overlapping detections and in most cases this increases false positives. While when we set the NMS threshold to a value close to 0, we lose any detection overlap with another one with a higher confidence score, which results in many false negatives.

3.3 Safety bar detection

3.3.1 Implementation details

We consider three objects for the safety bar detection task: open safety bar, closed safety bar, and background. In Faster R-CNN detection system, ZF and VGG16 pre-trained on ImageNet are used as a backbone to extract features, each in a dissociated experiment. The system is trained using back-propagation and the stochastic gradient descent algorithm in an end-to-end manner. The maximum number of epochs is set to 10000. The learning rate is set to 0.001, the learning rate decay to 10^{-5}, and the momentum to 0.9. At test time, only the object of highest confidence (softmax) score is considered (because it is impossible to have open and closed safety bar in the same image). NMS is applied with a threshold equal to 0.9. The dataset considered in the following experiments is the 2018 version of the chairlifts dataset represented in Section 1.2.3; we use the *large chairlifts dataset* unless otherwise specified.

3.3.2 Results

Table 3.2 summarizes safety bar detection results in the three settings *OOC*, *All*, and *LOCO* with ZF and VGG16 feature extractors. Figure 3.5 shows example detections on the test sets, with the corresponding ground truth boxes. All the examples in this figure are correct detections *i.e.* IoU >0.5, however, we notice that the softmax scores of the class 'open' are very low in multiple cases, indicating that the model is not confident about the existence of these objects even if the predicted class is correct. If we set a confidence threshold (*e.g.* 0.5), we will undoubtedly have many false negatives in this class. Figure 3.6 shows examples of false detections *i.e.* IoU <0.5; here, we notice that the sizes of the boxes are arbitrary, indicating a box regression problem. We also notice the problem of the softmax score being low for 'open' objects, even if the class is correct.

Table 3.2 – Open/closed safety bar detection results in the three settings *OOC*, *All*, and *LOCO* using Faster R-CNN model trained on the *large chairlifts dataset*, with ZF and VGG16 backbones pretrained on ImageNet. All the presented results are averaged over five folds.

setting	Backbone	class	precision	recall	F1	AP	mAP
OOC	ZF	closed	0.96	0.97	0.97	0.96	0.95
		open	0.97	0.96	0.96	0.94	
	VGG16	closed	0.99	0.99	0.99	0.99	0.99
		open	1.00	1.00	1.00	1.00	
All	ZF	closed	0.99	0.98	0.99	0.98	0.98
		open	1.00	0.99	1.00	0.99	
	VGG16	closed	0.99	0.99	0.99	0.99	0.99
		open	1.00	1.00	1.00	1.00	
LOCO	ZF	closed	0.71	0.77	0.72	0.68	0.59
		open	0.63	0.62	0.62	0.50	
	VGG16	closed	0.91	0.86	0.87	0.85	0.84
		open	0.89	0.87	0.88	0.84	

Tables A.1 - A.6 in Annex A show detailed results for each chairlift independently in the three settings.

OOC. In this setting, the dataset is small and with too few variations. Faster R-CNN produces 0.95 mAP in *OOC* setting using the ZF feature extractor, which implies that when trained on images belong to a chairlift the detector is able to detect with high precision and recall most of the objects in images from the same chairlift. The mAP achieved by Faster R-CNN when the very deep VGG16 model is used as a feature extractor increased to 0.99 (0.04 better than ZF feature extractor).

All. As expected the best results have been achieved in this setting, because of the huge size of the dataset on one hand, and all the domains are well-represented on the other hand with considerable variations. The mAP achieved by Faster R-CNN when VGG16 is used as a feature extractor slightly increased to 0.99 (0.01 better than ZF feature extractor).

LOCO. In *LOCO* experiment with ZF backbone, the performance of both classes decreased for all the measures, mAP for the different chairlifts has a wide range between 0.15 and 0.9, which is expected because in this setting, the images used in the test belongs to a chairlift completely unseen before. If the safety bar of this chairlift is similar to other safety bars well represented in the dataset, there is a good chance to have good results. Where if it is completely different, or similar to a safety bar which is not well represented in the dataset, that will harm the performance. These results underline the need to have training examples for all the chairlifts to achieve good results. Using VGG16 backbone in *LOCO* experiments reduce the harm in performance, with mAP equal to 0.84 (0.25 better than ZF backbone). The detector easily profited from the deeper and more expressive features of VGG16.

FS. The objective of this experiment is to train the model on 15 chairlifts and then study its ability to generalize on 6 randomly-selected chairlifts unseen before (C_0, C_1, C_2, C_6, C_9 and C_{12}). In this experiment, we use the *tiny chairlifts dataset* presented in Section 1.2.3 with only 300 training images, because the objective is to test the model generalizability when only a few training data is available.

Table 3.3 – Accuracy of Faster R-CNN and the baseline classifier with VGG16 backbone averaged over all target chairlifts in the *tiny chairlifts dataset* using the configuration *FS*.

Approach	classification accuracy
Image classification	85.71
Object detection	87.12

From this empirical study, we have found that object location information can improve image classification performance. The baseline classifier (trained using image-level annotations) gave an accuracy of 85.71%, while the object detector (trained and tested using instance-level annotations) achieved an accuracy of 87.12% Which means that the features extracted only from the regions that are most likely to contain the object, are better than features extracted from the entire image. Also, it is worth mentioning that the training bounding box annotations have been automatically obtained, as described in Chapter 1, Section 1.2.2.

3.4 People detection

3.4.1 Implementation details

For the task of people detection, we consider three objects in total : adult, child, and background. ZF and VGG16 pre-trained on ImageNet are used as backbones to extract features, each in a different experiment. The network is trained using back-propagation and the stochastic gradient descent algorithm in an end-to-end manner. The maximum number of epochs is set to 10000. The learning rate is set to 0.001, the learning rate decay to 10^{-5} and the momentum to 0.9. At test time, all objects with confidence (softmax) score higher than 0.5 are considered. NMS is applied with a threshold equal to 0.3. All the presented results are averaged over five folds. The dataset considered in the following experiments is 2019 version of the chairlifts dataset represented in Section 1.2.3.

3.4.2 Results

Table 3.4 summarizes people detection results in the three settings *OOC*, *All*, and *LOCO* with ZF and VGG16 feature extractors. Figure 3.7 shows example detections on test sets, with the corresponding ground truth boxes. Figure 3.8 show examples of false positives where an objecte is detected where there was no object, or the object is misclassified.

Tables A.7 - A.12 in Annex A show detailed results for each chairlift independently in the three settings.

As a consequence of the fact that children represent a small part of the totality of objects in the dataset (there are 12 times as many adults as children), it is difficult to interpret the global mAP value. To verify how the system performs if we ignore the distinction between adults and children, we conducted a simple experiment on the models trained on adult-child detection task in the three settings: at test time, we consider any object detected as 'adult' or 'child' belongs to a unified class and we call it 'person', and we consider true positive any detection with IoU ≥ 0.5 whatever the predicted class ('adult' or 'child'), and in the same manner a wrong detection is any proposal has IoU <0.5 with all the ground-truth boxes, also without taking into consideration the class name. Results represented in Table 3.5.

OOC. Faster R-CNN produces 0.76 mAP in *OOC* setting using both feature extractors ZF and VGG16, 0.54 for 'child' class, and 0.98 for 'adult' class. This means that the detector is able

Table 3.4 – Adult/child detection results in the three settings *OOC*, *All* and *LOCO*, using Faster R-CNN model with ZF and VGG16 backbones pretrained on ImageNet.

setting	Backbone	class	precision	recall	F1	AP	mAP
OOC	ZF	child	0.87	0.57	0.66	0.54	0.76
		adult	0.95	0.98	0.96	0.98	
	VGG16	child	0.91	0.55	0.67	0.54	0.76
		adult	0.95	0.98	0.97	0.97	
All	ZF	child	0.90	0.36	0.50	0.34	0.66
		adult	0.90	0.98	0.94	0.97	
	VGG16	child	0.91	0.56	0.69	0.54	0.76
		adult	0.93	0.99	0.96	0.98	
LOCO	ZF	child	0.83	0.31	0.42	0.29	0.61
		adult	0.90	0.95	0.91	0.93	
	VGG16	child	0.91	0.45	0.58	0.43	0.70
		adult	0.92	0.98	0.95	0.97	

Table 3.5 – Person detection results in the three settings *OOC*, *All* and *LOCO*, using Faster R-CNN model with ZF and VGG16 backbones pretrained on ImageNet.

setting	Backbone	precision	recall	F1	AP
OOC	ZF	0.95	0.99	0.96	0.99
	VGG16	0.96	0.99	0.98	0.99
All	ZF	0.93	0.98	0.95	0.97
	VGG16	0.95	0.99	0.97	0.99
LOCO	ZF	0.92	0.94	0.93	0.94
	VGG16	0.94	0.98	0.96	0.98

to detect with high precision and recall the 'adult' objects even if the dataset is small. However, many 'child' objects are left behind without being detected or detected as 'adult', which led to a low recall, which in turn led to poor AP. However, the precision for 'child' class is better than the recall because most of the objects detected as 'child' actually belong to the 'child' category while few of them actually belong to the 'adult' category, that is why 'adult' precision is slightly lower than its recall. The mAP was not noticeably increased when the very deep VGG16 model used as a feature extractor. Looking at the performance of the detector in 'person' detection, we notice that it is very good in terms of AP using ZF or VGG16 backbone. This implies that the proposals are good but not the classification. However, the detection of children is important to ensure their safety, so this is a weakness that needs to be addressed.

All. Contrary to safety bar detection, no better result was obtained in this setting, because people objects are not chairlift-dependent, so the enormous size of the dataset did not add extra variation or make the objects more well-represented. Instead Faster R-CNN with ZF backbone has lower performance than *OOC* setting. The mAP achieved by Faster R-CNN when VGG16 used as feature extractor increased to 0.76 (0.1 better than ZF feature extractor). In this experiment in general it is worthy to use deeper network as backbone, but it is not better than using any backbone in *OOC* setting. But given the results of detecting 'person', VGG16 is better than ZF (+0.2 of AP), and both are significantly better in this task than the task of 'adult'/'child' detection.

LOCO. In *LOCO* experiment with ZF backbone, the performance of 'child' classes decreased for all the measures, mAP for the different chairlifts has a range between 0 and 0.51. Using VGG16 backbone in *LOCO* experiments reduces harm in performance, with mAP equal to 0.7 (0.09 better than ZF backbone). The detector profited from the deeper and more expressive features of VGG16. For 'adult' class the performance is equivalent to the other settings where test images belong to a chairlift used for training too, because as mentioned before people objects are not dependent on the chairlift. Considering the detection results of 'person', VGG16 is better than ZF (+0.4 of AP), and both are significantly better than 'adult'/'child' detection.

3.5 Conclusion

From the experiments conducted in this chapter, we conclude the following: there is a strong need for a solution to improve the generalization across chairlifts, we will address this problem by domain adaptation in Chapter 3, and by a generalization boosting method in Chapter 6.

Learning how to correctly locate the object in the image is very promising to improve the accuracy of classification. However, this requires costly instance-level bounding box annotations, so it is necessary to find a way to help in the training of the network when few labeled data is available, as we will see later in Chapter 5 we propose to solve this issue by inserting shape priors.

Person detection results indicate that a chairlift-dependent model is preferable for distinguishing children from adults. This is probably due to the fact that the use of a chairlift-specific model allows that model to learn specific features indicating the presence of children. One can think of vehicle elements that are only visible when the passenger is a child. These elements are not the same if the geometry of the chairlift is changed, and therefore they are not correctly learned in the setting *All*. *LOCO* results support this interpretation: children are identified according to the geometry of the chairlift and an unknown geometry leads to lower performance.

One perspective is to evaluate the detection of people and safety bars in a unified framework (which was not possible because we did not have annotated version of the dataset with all classes).

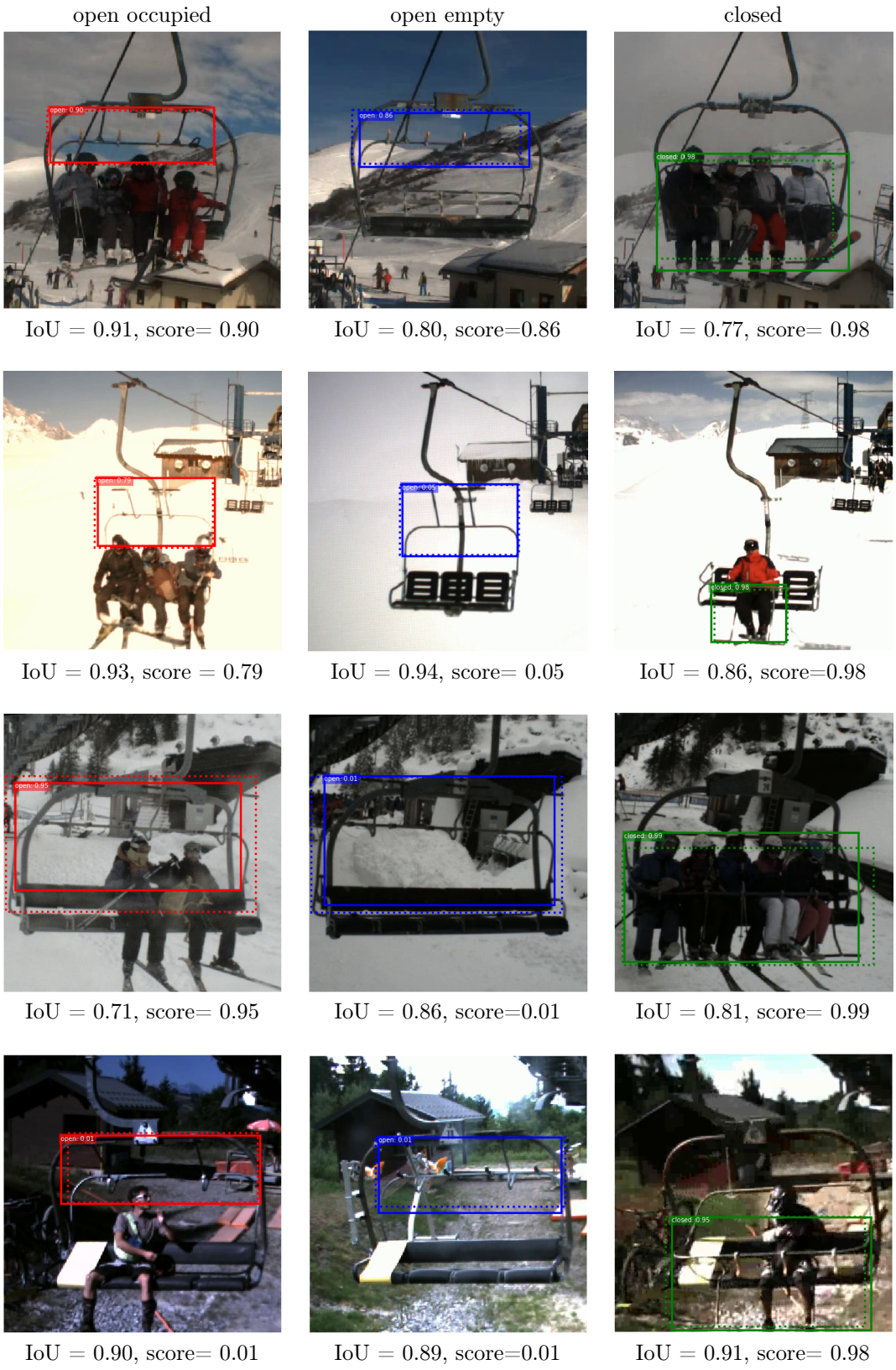


Figure 3.5 – Selected examples of open/closed safety bar object detection results using Faster R-CNN on the 2018 chairlifts dataset. The dotted line represents the ground truth and the solid line represents the detection. Under each image, we provide IoU between the ground truth bounding box and the predicted one, and the softmax score $\in$ [0,1]. All examples in this figure are correct detections. Chairlifts by line are, respectively, C_1, C_2, C_{19} and C_{20}.

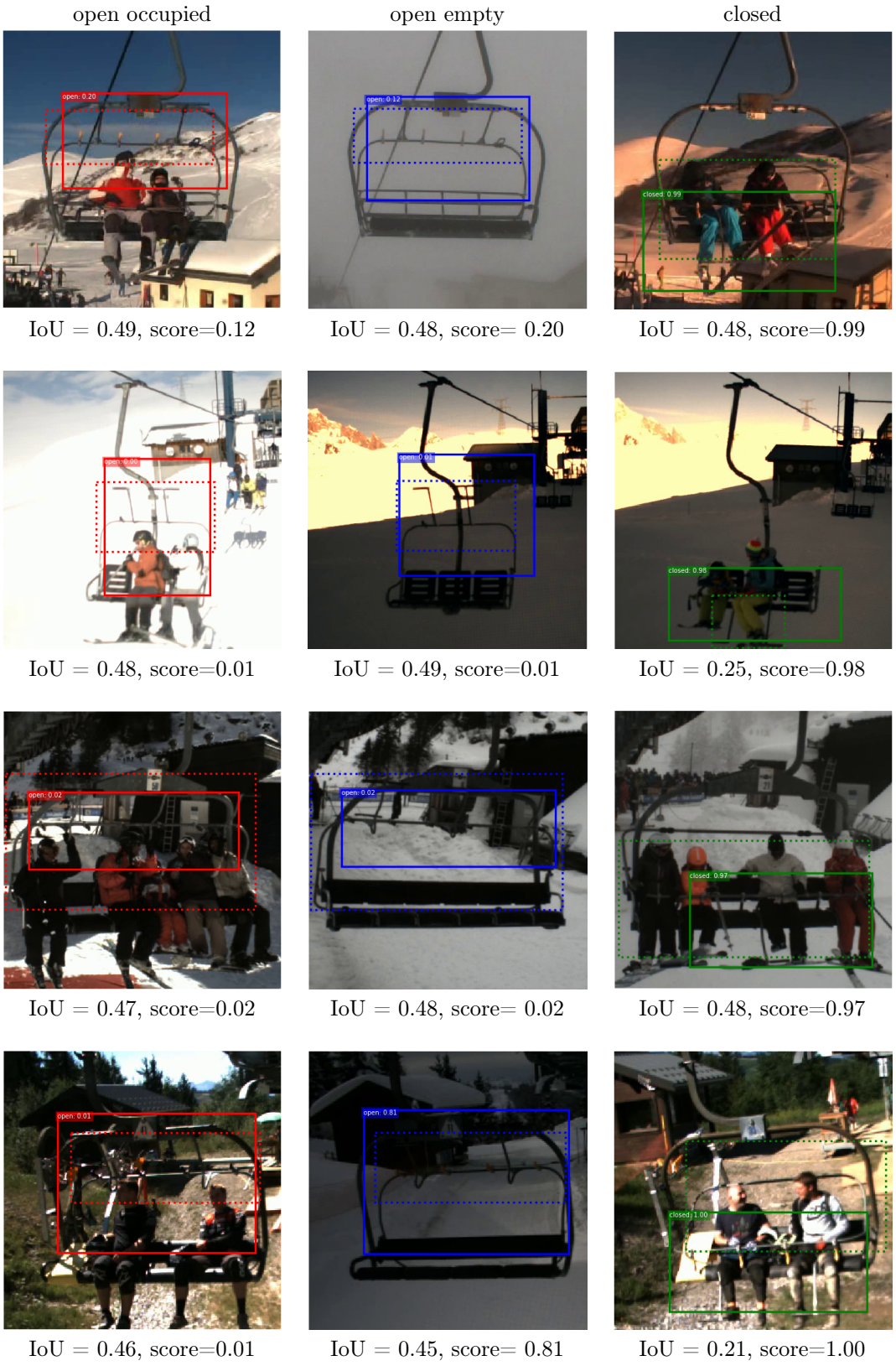
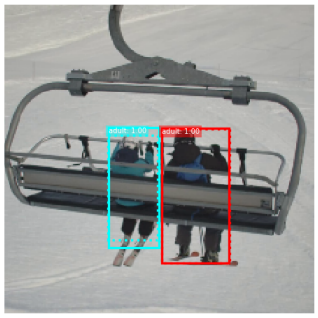
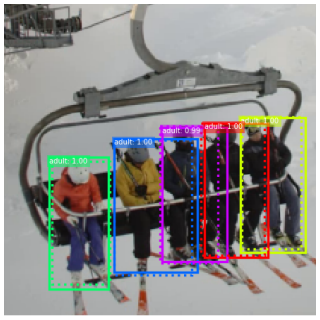


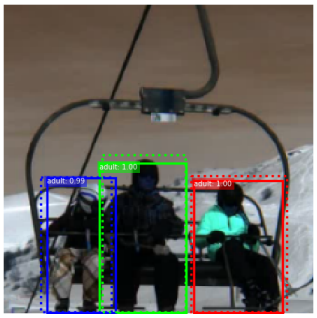
Figure 3.6 – Selected examples of open/closed safety bar object detection results using Faster R-CNN on 2018 chairlifts dataset. The dotted line represents the ground truth and the solid line represents the detection. Under each image we provide IoU between the ground truth bounding box and the predicted one, and the softmax score $\in$ [0,1]. All examples in this figure are false detections. Chairlifts by line are, respectively, C_1, C_2, C_{19} and C_{20}.

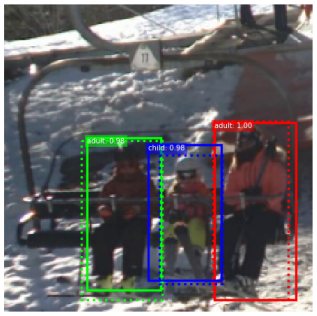

adult (1), child (1)


adult (1), adult (1)

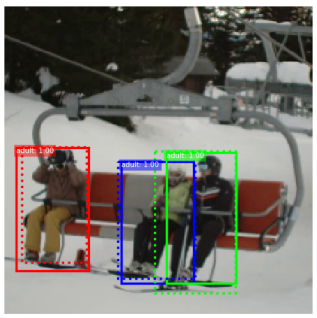

adult (1), adult (1), adult (1)
adult (0.99),adult (1)

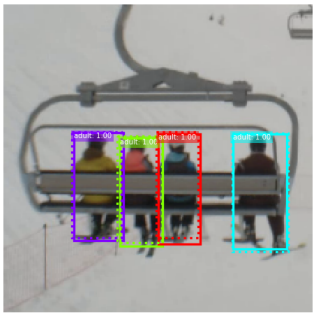

adult (1), adult (1)

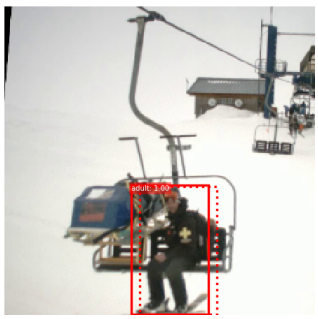

adult (0.99), adult (1)
adult (1)

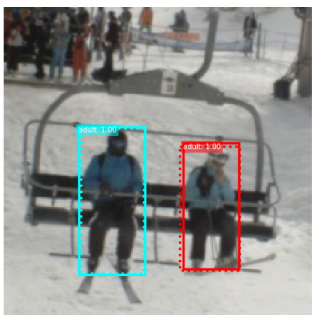

adult (0.98), child (0.98),
adult (1)


adult (1)


adult (1), adult (1)
adult (1)


adult (1), adult (1),
adult (1), adult (1)


adult (1)


adult (1), adult (1)


adult (1), adult (1), adult (1),
adult (0.99), adult (1)

Figure 3.7 – Selected examples of adult/child object detection results using Faster R-CNN on 2019 chairlifts dataset. The dotted line represents the ground truth and the solid line represents the detection, the number in the top left corner of each box (also in parenthesis under each image) represents the softmax score $\in$ [0,1], next to the detected class name.

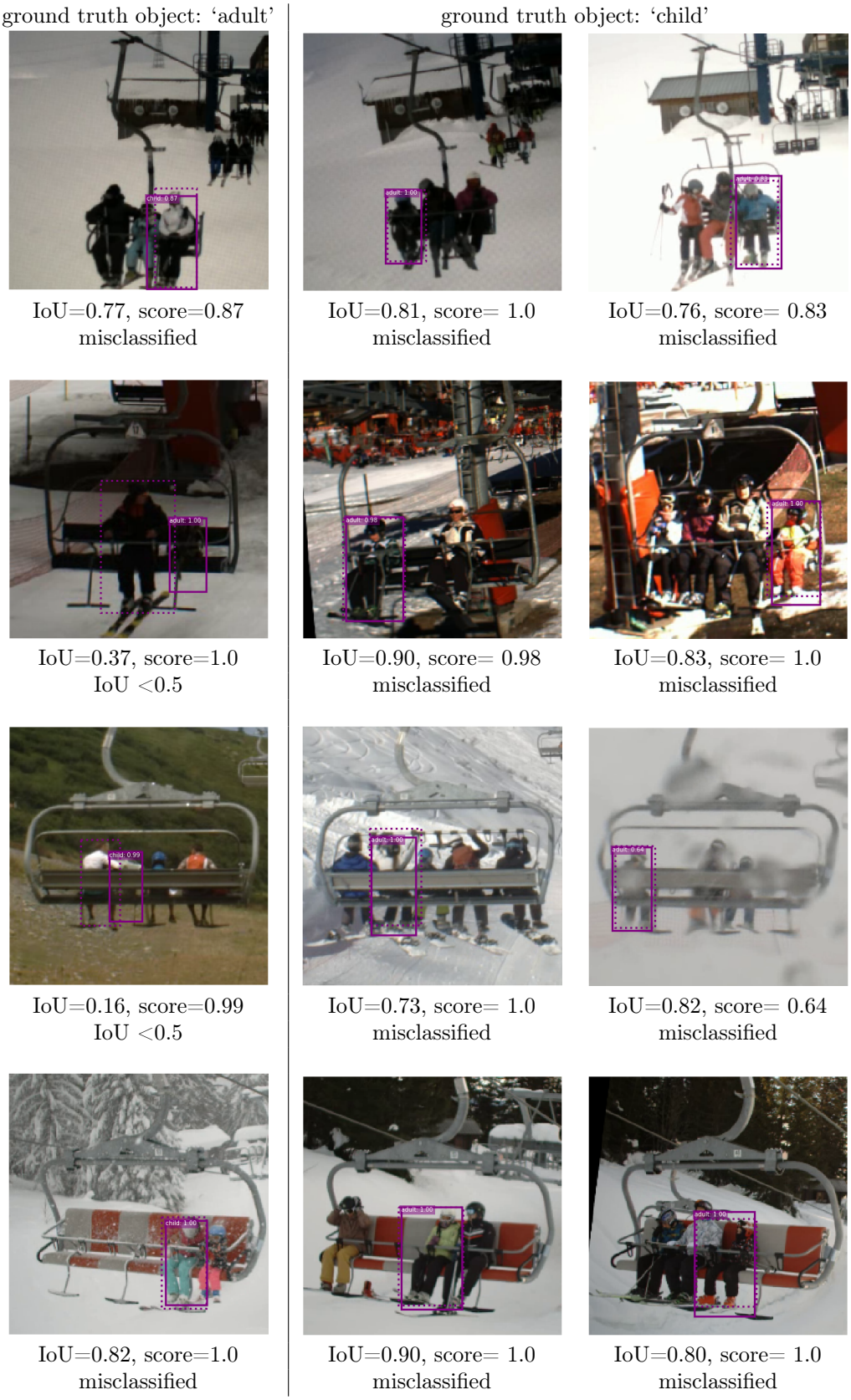


Figure 3.8 – Selected examples of adult/child object detection results using Faster R-CNN on 2019 chairlifts dataset. The dotted line represents the ground truth and the solid line represents the detection. All examples in this figure are false detections. Under each image we provide the softmax score $\in$ [0,1], and the highest IoU between a ground truth bounding box and the predicted one.

Chapter 4

Domain Adaptive Object Detection

The experiments in Chapter 3 have demonstrated a vulnerability of Faster R-CNN when there is a domain shift between the training and test data. The models learned on some chairlifts tend not to generalize well to new unseen chairlifts. In this chapter, we present a new viewpoint about the domain shift problem in object detection. Furthermore, we propose to adapt the Region Proposal Network (RPN) and integrate this new adaptation module in Faster R-CNN. Appropriately, we run experiments in two different application contexts: autonomous driving and chairlift safety problem. This work is presented in a paper [37] in the *International Conference on Advanced Concepts for Intelligent Vision Systems, 2020.*

4.1 Introduction

As previously stated in Chaper 1 Section 1.1, Mivao project aims to develop a computer vision system that acquires images from the boarding station of chairlifts, analyzes the important elements *e.g.* people, chairlift carrier, safety bar, *etc.* and triggers an alarm in case of dangerous situations. In Chapter 3, we proposed to identify dangerous situations in chairlifts by detecting the important elements in the scene *i.e.* people and safety bar, using an object detection framework based on deep learning. We have experimentally shown that this approach can provide better generalization on new unseen chairlifts than classical classification methods at the cost of having instance-level box annotations. To go a step further, in this chapter, we investigate the use of domain adaptation components to provide even better results in new unseen chairlifts.

As discussed in Chapter 2, the research field of domain adaptation is very active in the computer vision community because standard deep learning methods require large training datasets with instance-level annotations. However, few annotations are available for most real-world applications due to the lack of image sources, copyright issues, or annotation costs. For example in our chairlift safety problem, we have a large dataset provided by Bluecime we managed to annotate it automatically with safety bar bounding boxes, and through crowdsourcing with people bounding boxes. However, it is so expensive to annotate new images every time a new chairlift enters the system.

To overcome this problem, a current trend consists in training the network on a large annotated dataset (source domain) *i.e.* the chairlifts annotated dataset in our case; while adapting the network features to the tested dataset (target domain) *i.e.* non-annotated images from new chairlifts. This approach is called domain adaptation [99, 101]. Domain adaptation is needed when there is a distribution discrepancy between the source domain on which the model has been trained and the target domain for which few or no annotations are provided. In this chapter, we address this domain adaptation problem in the context of object detection in the case where no annotations are available in the target domain, which is called unsupervised domain adaptation (see Chapter 2 for more details). In this context, the case of autonomous driving has been extensively addressed, and a variety of datasets exists covering different urban scenes situations, illumination, and weather conditions [144, 145]. Therefore, we consider this scenario since our contribution is not limited to the chairlift safety problem.

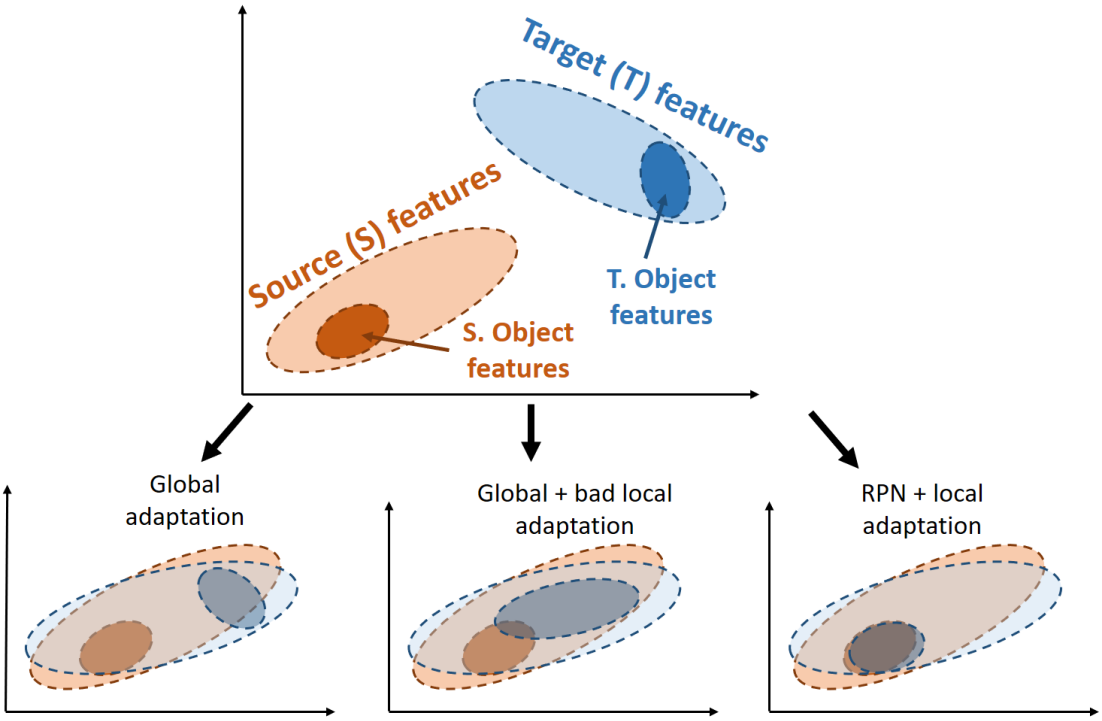


Figure 4.1 – Illustration of global, local and RPN adaptation (see text for details).

Traditionally, domain adaptation methods in the literature have been applied to image classification tasks, and there is relatively little work dealing with domain adaptation related to object detection, most notably in the unsupervised setting. Existing approaches added adversarial training components in the classical Faster R-CNN detector, at both global and instance levels but without adapting the Region Proposal Network (RPN), leading to a residual domain shift. A global adaptation is illustrated in (bottom left of Figure 4.1), which may not match source and target object features. Since the RPN is trained on the source domain, the proposals from the target images may be wrongly detected, and the local features used for the adaptation may be outside the target object features set (bottom center in Figure 4.1). In this chapter, we propose to adapt the RPN in order to ensure that the features extracted from the target images to overlap with the source object features. A local adaptation through adversarial learning will thus better align source and domain features (bottom right in Figure 4.1).

4.2 Related work

Object detection. The first approaches proposed in the context of object detection using CNN were based on the region pooling principle [146, 61]. In R-CNN [61], candidate regions detected by the selective search were represented by a subset of pooled features and evaluated by an instance classifier. This two-stage principle was further refined in Faster R-CNN [3] with a common CNN backbone to extract the whole image features and two different sub-pipelines: the first one called Region Proposal Network (RPN) to generate proposals of regions which are likely to contain objects, and the second one which is basically a classification and regression network aimed at refining the location and size of the object and finding its class. Besides these two-stage approaches, one-stage approaches directly predict box location, size and class in a single pipeline either by using anchor boxes with different aspect ratios [68] or by solving a regression problem on the feature grid [67]. Interested readers may refer to Section 2.3 for more details. Since Faster R-CNN [3] provides very accurate results and has been largely studied, especially for domain adaptive object detection [107, 108, 109, 111], we propose to consider this network as a baseline in our approach.

Domain adaptation. Unsupervised domain adaptation is needed when we want to learn a predictor in a target domain without any annotated training samples in this domain [99, 101]. Obviously, annotations are available in a source domain, which is supposed to be close to the target one. For example, in the chairlift safety problem, we train models using the annotated images from chairlifts, which we call the source domain. On the other hand, when we acquire non-annotated images from a new chairlift, referred to as the target domain, we cannot use the previously trained models directly on the target domain, because although it is close to the source domain, there is a domain shift in the shape of the vehicle, the number of seats, the viewpoint, *etc*.

Two main types of methods have been proposed in this context. The first one is to try to match the feature distribution in the source and target domains either by finding a transformation between the domains [147] or by directly adapting the features [148]. One noticeable example is the Gradient Reversal Layer (GRL) approach proposed by Ganin *et al*. [100] that attempts to match source and target feature distributions. They propose to jointly optimize the class predictor and the source-target domain disparity by back-propagation. The second type of methods relies on Generative Adversarial Networks (GANs) [97]. The principle is to generate annotated synthetic target images from the source images and to learn (or fine-tune) the network on these synthetic target data [105]. For more information, refer to Section 2.4.

Domain adaptation for object detection. Few works consider domain adaptation for object detection, particularly in the unsupervised setting. [104] proposes class-specific subspace alignment to adapt R-CNN [61], and [106] uses adversarial training inspired by [100] to adjust features at two different levels of Faster R-CNN architecture. The adaptation at image-level intends to eliminate the domain distribution discrepancy at the output of the backbone network, while the instance-level adaptation concerns the features which are pooled from a Region of Interest (RoI) before the final category classifiers. Following the same adversarial training approach, Saito *et al*. [108] argue that a global matching may adversely affect the performance for large domain shifts. They thus propose to combine a strong alignment of local features and a weak alignment of global ones. Shan *et al*. [109] propose to minimize domain shift at both the pixel and feature level. In this approach, the features could be invariant in the domain, but the proposals are not adapted, resulting in limited localization capability. Hsu *et al*. [111] propose to break down the adaptation into two steps that both solve an easier problem with a smaller domain gap. First, the source images are transformed to match the appearance of the target using CycleGAN image-to-image translation network. Next, adversarial learning is employed to align the resulting synthetic images with the target domain at the feature level.

To the best of our knowledge, none of the previous works considers the adaptation of the region proposal sub-network of Faster R-CNN. They are then sensitive to any shift in the distribution of object bounding boxes between source and target domains. More details can be found in Section 2.4.2.

In this work, we propose to incorporate two adversarial domain adaptation modules in Faster R-CNN: the first one at RPN-level to address the source-target domain shift of features of the region proposal module, and the second one at instance-level to adapt the RoI-pooled features used in the final classification module.

4.3 Region Proposal Oriented approach

In order to explain our adaptation scheme, we have to review in detail the workflow of Faster R-CNN [63], depicted in Figure 4.2. Then, we present our approach to adapt this detector between different domains.

4.3.1 Faster R-CNN

Faster R-CNN is basically composed of two convolutional blocks called C_1 and C_2, providing two feature maps F_1 and F_2, respectively (see Figure 4.2). Based on F_2, the RPN predicts a set of box positions used to crop the feature map F_1 using the RoI pooling layer (called RP layer, hereafter). It is worth mentioning that the gradient can not be back-propagated through the RP layer towards the RPN because this step is not differentiable. The authors of Faster R-CNN resort to an alternating training to cope with this problem [63]. It is crucial to understand this point when one wants to apply domain adaptation to Faster R-CNN. It means that we can not

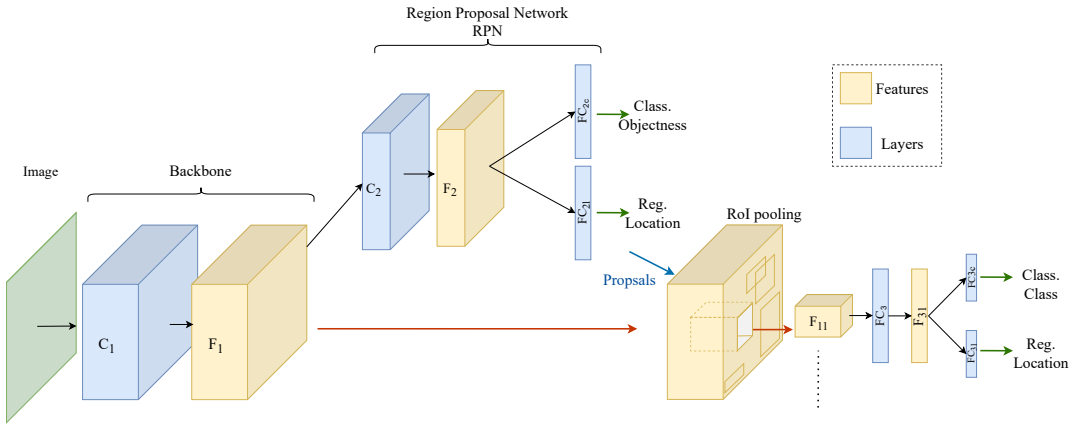


Figure 4.2 – Faster R-CNN Workflow.

just plug a domain adaptation module after the last layers of Faster R-CNN (namely F_{3i}) and adapt in one shot the classification layers and the convolution blocks C_1 and C_2.

Back to the workflow of Faster R-CNN, the outputs F_{1i}, $i = 1, ..., N_p$, of the RP layer are cropped and resized parts of the feature map F_1 . N_p is the number of proposals returned by the RPN. The feature maps F_{1i} are then sent to the shared fully connected layers FC_3 whose outputs F_{3i} are used to make the final decision of class and location.

From this workflow, we note that the classification and regression layers take as inputs either F_2 or F_{3i}, which are the key feature maps of the detector. In the next section, we present how these feature maps can be adapted between the two domains.

4.3.2 Adapting Faster R-CNN

Let us consider a source domain $\mathcal{S}$ with $N_\mathcal{S}$ images $\{I_i^\mathcal{S}\}$, $i = 1, ..., N_\mathcal{S}$, each containing $n_i^\mathcal{S}$ objects, located at the positions $l_{ij}^\mathcal{S}$ and associated with the classes $c_{ij}^\mathcal{S}$, $j = 1, ..., n_i^\mathcal{S}$. Likewise, we denote $\mathcal{T}$ a target domain constituted of $N_\mathcal{T}$ target images $\{I_i^\mathcal{T}\}$, $i = 1, ..., N_\mathcal{T}$, each containing $n_i^\mathcal{T}$ objects, located at the positions $l_{ij}^\mathcal{T}$ and associated with the classes $c_{ij}^\mathcal{T}$, $j = 1, ..., n_i^\mathcal{T}$.

If the two domains are different (cameras, viewpoints, weather conditions,...), there is a domain shift between the joint distributions $P(I^\mathcal{S}, l^\mathcal{S}, c^\mathcal{S})$ and $P(I^\mathcal{T}, l^\mathcal{T}, c^\mathcal{T})$. In this case, we can not train the detector on the source data and obtain good results on the target data, without adaptation. The aim of domain adaptation is to decrease this distribution discrepancy so that $P(I^\mathcal{S}, l^\mathcal{S}, c^\mathcal{S}) \approx P(I^\mathcal{T}, l^\mathcal{T}, c^\mathcal{T})$. In the context of unsupervised domain adaptation, the labels (locations and classes) of the target data are not available and this is not an easy task to decrease the joint distribution discrepancy. By applying the Bayes' rule on the joint distribution, we obtain, for the source domain:

$$P(I^\mathcal{S}, l^\mathcal{S}, c^\mathcal{S}) = P(l^\mathcal{S}, c^\mathcal{S}|I^\mathcal{S})P(I^\mathcal{S}) \tag{4.1}$$

Most of domain adaptation approaches assume a covariate shift, which means that the shift between the source and target joint distributions is caused by the marginal distributions $P(I)$, while the conditional distributions $P(l, c|I)$ are constant across domains, *i.e.* $P(l^\mathcal{S}, c^\mathcal{S}|I^\mathcal{S}) = P(l^\mathcal{T}, c^\mathcal{T}|I^\mathcal{T})$. Under this assumption, in order to decrease the joint distribution discrepancy, we have just to decrease the marginal distribution shift, so that $P(I^\mathcal{S}) \approx P(I^\mathcal{T})$. In order to change the marginal distributions of the images, the classical approaches apply a transform T on the image features, so that $P(T(I^\mathcal{S})) \approx P(T(I^\mathcal{T}))$. Usually, the transform T is a part of a convolution neural network.

In this approach, we propose to consider and adapt different feature maps extracted from the images. By looking at Figure 4.2, we note that two feature maps are used as input for classification and regression layers, namely the feature map F_2 and the feature vectors F_{3i}. So, in order to adapt the detector to the source domain, we have to adapt the marginal distributions of F_2 and F_{3i}, so that $P(F_2^\mathcal{S}) \approx P(F_2^\mathcal{T})$ and $P(F_{3i}^\mathcal{S}) \approx P(F_{3i}^\mathcal{T})$.

In order to enforce these distributions to be closer, we propose to resort to an adversarial domain adaptation approach [100] called gradient reversal layer (GRL). Note that any other

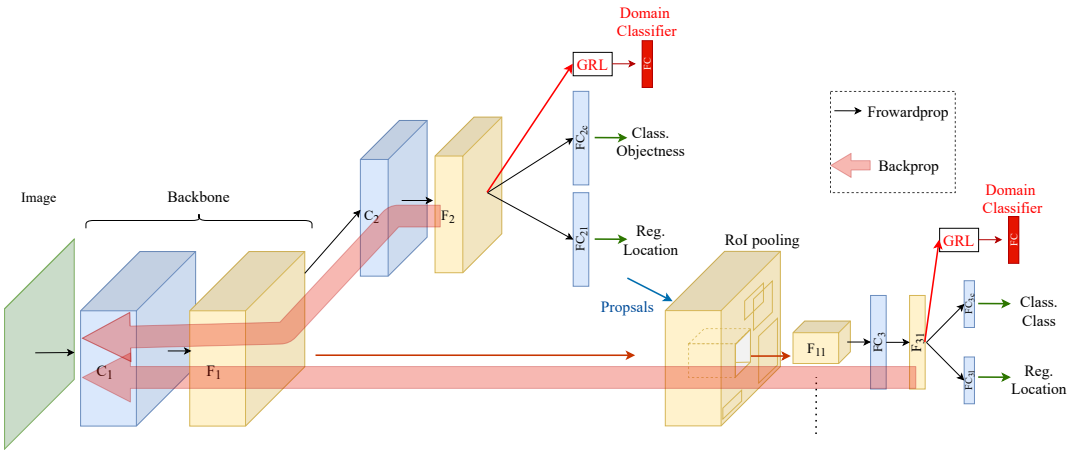


Figure 4.3 – Our domain adaptation for Faster R-CNN. See text for details.

adversarial domain adaptation algorithms could have been used, we just use this one for a fair comparison with [106]. When plugged on a feature map F_k, the idea of GRL is to minimize the discrepancy between the feature distributions over the source and target domains $P(F_k^{\mathcal{S}})$ and $P(F_k^{\mathcal{T}})$ [100]. If the GRL is able to perfectly overlap these two distributions, we can conclude that the features extracted at this point of the network (F_k) are domain invariant and so can be used either on the source or target domain with equivalent accuracies.

From the previous analysis, it is obvious that two GRL modules should be inserted in the detector: one after the feature map F_2 and one after the feature vector F_{3i}. It is worth mentioning that, when we plug a GRL module to a feature map, we back-propagate the (reverse-)gradient until the first layer of the convolutional block C_1. Thus, the main advantage of our approach is that the reversal gradients are back-propagated through all the layers of the detector. Consequently, the backbone, the RPN and the local features are all adapted (see Figure 4.3).

Formally, at training time, the total loss corresponding to a given training image $I_k \in I^{\mathcal{S}} \cup I^{\mathcal{T}}$ from domain $d_k \in \{\mathcal{S}, \mathcal{T}\}$ is given by:

$$L = L_{Fst} - \lambda \sum_{i,j} L_H \left(FC_{2a}(F_2^{i,j}(I_k)), d_k \right) - \lambda \sum_{i=1}^{N_p} L_H \left(FC_{3a}(F_{3i}(I_k)), d_k \right) \tag{4.2}$$

where L_{Fst} denotes the original Faster R-CNN loss activated only if $I_k \in I^{\mathcal{S}}$, L_H denotes the cross-entropy loss, λ denotes the trade-off parameter to balance Faster R-CNN loss and domain adaptation losses, FC_{2a} and FC_{3a} denote the fully connected predictors for domain adaptation, $F_2^{i,j}(I_k)$ denotes the feature vector at location (i, j) of feature map F_2 for image I_k, and $F_{3i}(I_k)$ denotes the feature vector corresponding to the proposal region i of image I_k.

We note that the recent domain adaptive detection approaches [106, 108] have not tried to adapt the RPN layer, and we think that this is a strong weakness of these approaches. Indeed, as mentioned in [106] (called DA-Faster hereafter), the image-level adaptation is enforcing the target and source feature distributions F_1 to be closer but it is very hard to perfectly align them. This is one of the reasons why DA-Faster approach also applies instance-level adaptation. But, it is clear in Figure 4.2, that if the features F_1 are not well adapted between the domains, the output of the RPN will also be different between the domains and consequently, the locations where the boxes F_{1i} are cropped from F_1 will be domain-dependent. Therefore, the instance-level adaptation on F_{3i} features will not help to adapt the object detector between domains, since it will work on local features which are not equivalent between the domains (see Figure 4.1).

4.4 Experiments

The purpose of our domain adaption solution was to improve the detection accuracy of the safety bar in a new chairlift with a model trained on another one. So we propose to test our algorithm on the chairlift safety problem. However, in order to be able to compare with existing approaches on referenced benchmarks, we also propose to first apply our solution to another interesting problem: autonomous driving.

4.4.1 Experimental setup

In these experiments, we train on a source dataset and test on a target dataset from a different domain. During training, likewise the other domain adaptive approaches, we also use images from the target domain, but without any label, while the source dataset images are provided with their bounding boxes instance annotations.

The baseline is Faster R-CNN model trained only on the source dataset. As mentioned earlier, our solution is inspired by DA-Faster [106] but our contribution is in the analysis of the domain shift in Faster R-CNN, conducting to the solution that the domain adaptation module (GRL) should be plugged at RPN-level. Consequently, the aim of these experiments is to compare DA-Faster with our approach in order to check the validity of our contribution in practice. Thus, for all the experiments, we compare our approach with DA-Faster [106]. As mentioned in [108], the results provided by the authors of DA-Faster are unstable and Saito *et al.* proposed to re-implement their own code for DA-Faster, conducting to lower results than the original paper [106]. So like [108], we report the results of DA-Faster with the implementation provided by [149] with the same hyper-parameters as our solution (results denoted *DA-Faster* hereafter), as well as the results provided by the original paper [106] (denoted *DA-Faster**), when available on the considered dataset.

To evaluate object detection, we report the mean Average Precision (mAP) with intersection over union (IoU) threshold at 0.5 (denoted AP50), the mAP with IoU threshold of 0.75 (AP75) and the mAP averaged over multiple IoU from 0.5 to 0.95 with a step size of 0.05 (APcoco). The network is trained in an end-to-end manner using back-propagation and the stochastic gradient descent (SGD) algorithm. As a standard practice, Faster R-CNN backbone is initialized with pre-trained weights on ImageNet classification. We use a learning rate of 0.001 for 50k iterations, and 0.0001 for the next 20k iterations. Each iteration has two mini-batches, one from the source domain and the other from the target domain. The trade-off parameter λ to balance Faster R-CNN loss and domain adaptation loss is set to 0.1 as in [106]. We use momentum of 0.9 and a weight decay of 0.0005.

4.4.2 Autonomous driving

In this context, we evaluate the domain adaptive detectors for two domain shifts: weather conditions (foggy and not foggy) and acquisition conditions (different cameras, different viewpoints, and different scenes).

Figure 4.4 – One image from each dataset: the Cityscapes dataset (top left), its foggy version (top right) and KITTI dataset (bottom).

Cityscapes → Foggy Cityscapes. In the first experiment, we use Cityscapes dataset

[144] as a source domain. It is an urban scene dataset with 2975 training images and 500 validation images. The 1525 unlabeled test images are not considered. For training the network, we are using the 2975 training images and do not consider the validation images. There are 8 categories with instance annotations in this dataset, namely *person, rider, car, truck, bus, train, motorcycle* and *bicycle*. The target domain is Foggy Cityscapes [150] dataset generated by applying fog synthesis on Cityscapes dataset to simulate fog on real scenes (see Figure 4.4). Thus, the number of images and labels are exactly the same as for Cityscapes dataset. For testing the detection, we are using the 500 validation images from Foggy Cityscapes. The results are summarized in Table 4.1. First, we can note that, without domain adaptation, the results of Faster R-CNN are very bad, underlying the strong need for adapting the network in case of weather condition variations. Thus, DA-Faster improves the results over Faster R-CNN, but we note that our approach clearly outperforms DA-Faster on this dataset, showing that RPN adaptation helps in adapting the detector in case of weather condition variations.

Table 4.1 – Detection results on Foggy Cityscapes (trained on Cityscapes dataset). The AP50 is reported for each class as well as the average APcoco, AP50 and AP75 over all classes.

Method	person	rider	car	truck	bus	train	mcycle	bicycle	APcoco	AP50	AP75
Faster R-CNN	18.8	20.5	24.2	17.0	08.0	06.2	07.2	05.0	06.20	13.35	05.42
DA Faster R-CNN	27.3	35.7	44.1	20.3	35.2	8.9	16.2	23.6	12.28	26.41	10.02
DA Faster R-CNN*	25.0	31.0	40.5	22.1	35.3	20.2	**20.0**	27.1	-	27.60	-
Ours	**27.8**	**35.8**	**45.1**	**23.5**	**42.1**	**26.1**	18.0	**27.6**	**13.70**	**30.47**	**11.04**

Cityscapes → KITTI. In this experiment Cityscapes is the source domain, and KITTI [145] is the target domain (see Figure 4.4). KITTI is a benchmark for autonomous driving, which consists of 7481 training images. Since the test set is not annotated, we use all the training images at test time to evaluate the performance. Only one category (*car*) is annotated in KITTI, so we consider this single class for evaluation. The results are summarized in Table 4.2. Once again, we note that the domain adaptation helps improving Faster R-CNN results. We see also that our approach outperforms DA-Faster for all the criteria when using the same hyperparameters. The results provided in [106] are better than ours for $AP50$, but note that the implementation and hyper parameters are different from our tests. The comparison is therefore not fair.

Table 4.2 – Detection results on KITTI training set (trained in Cityscapes dataset) for one class (Car) detection.

Method	APcoco	AP50	AP75
Faster R-CNN	26.73	58.60	21.54
DA Faster R-CNN	27.51	60.38	22.67
DA Faster R-CNN*	-	**64.10**	-
Ours	**28.39**	61.32	**23.59**

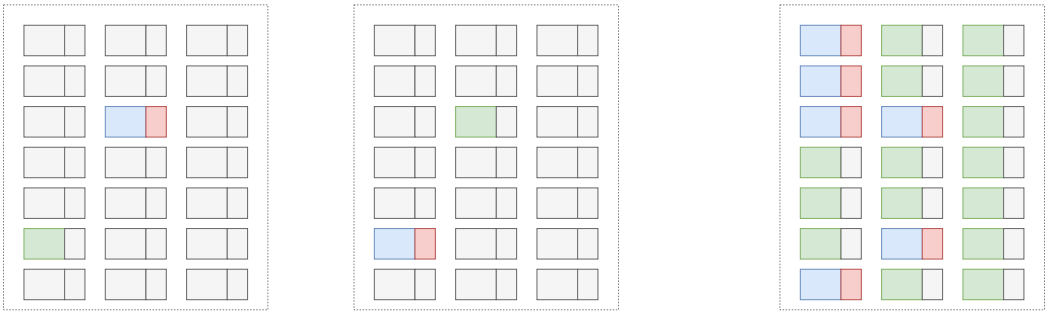
4.4.3 Chairlift safety problem

In this experiment, we tackle the chairlift safety problem as an object detection task trying to detect the safety bar in the image, considering that it has to be closed when the chairlift leaves the boarding station. Across the ski resorts, the viewpoint, the background, the carrier geometry, and the camera may be different; and domain adaptive detectors are required to install the system for new chairlifts without a fastidious and time-consuming step of manual annotation. To simulate that we conducted the experiments in the following two settings.

Chairlift vs. chairlift

Chairlifts dataset. For this experiment, we consider one chairlift as the source domain and another one as the target domain (see Figures 4.5a, 4.5b). As it is too cumbersome to make all

the combinations between all the chairlifts, we have randomly selected two different chairlifts (C_9 and C_6), called hereafter chairlift1 and chairlift2, and we took the corresponding training and test images from the *large chairlifts dataset*. Example images are provided in Figure 4.6. We can note that the main differences between the two chairlifts are in the viewpoints which are slightly different and in the presence of a cluttered background in the chairlift2. The images are centered on the chairlift and manually labeled with image-level annotations. Instance annotations has been created automatically as described in Section 1.2.2, with two categories: *open safety bar* and *closed safety bar*.



(a) chairlift1 → chairlift2 (b) chairlift2 → chairlift1 (c) *FS*

Figure 4.5 – Chairlift vs. chairlift, and *FS* settings. Green: source labeled training data, blue: target unlabeled training data, and red: target test data.

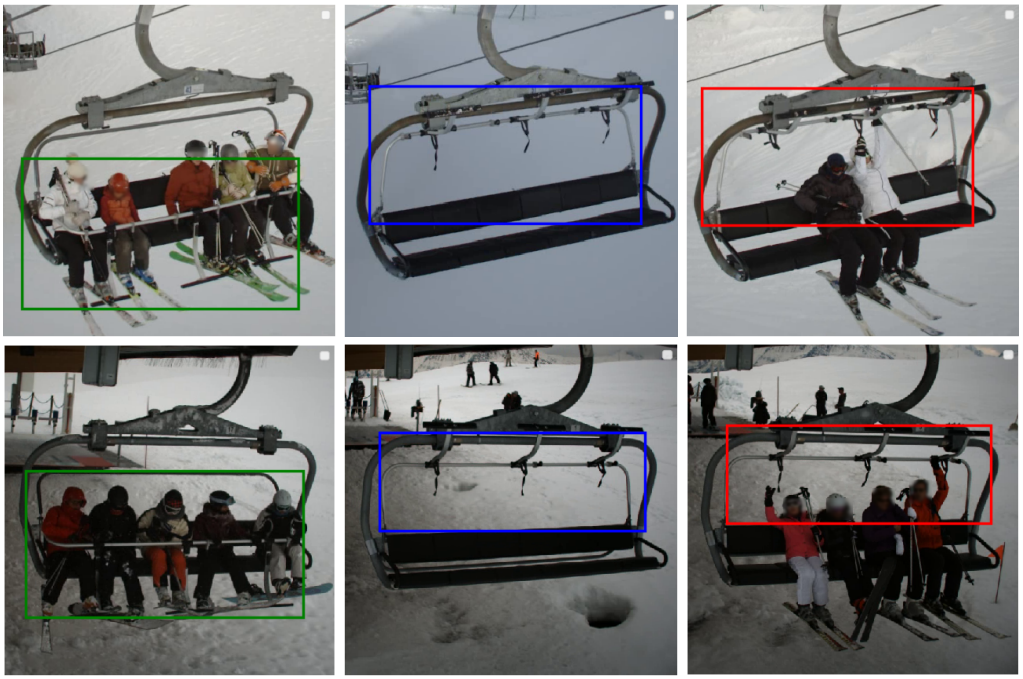


Figure 4.6 – Example images from chairlift1 (top) and chairlift2 (bottom). The box annotations (open:red or blue, and closed:green) are provided for illustration.

Evaluation. The results are provided in Table 4.3. By training the baseline Faster R-CNN using images from one chairlift and test it on images from another chairlift, the results were surprisingly very good in terms of AP50. This can be explained by the important size of the ground truth bounding boxes that have a high chance to well overlap random bounding boxes with similar dimensions. Obviously, when looking at the more demanding criteria such as AP_{coco} or $AP75$, the need for domain adaptation is evident for precise object detection. The results show that the two domain adaptive detectors (DA-Faster and ours) are equivalent for the adaptation from chairlift1 to chairlift2, but they also show that our adaptation is much better than DA-Faster for the adaptation from chairlift2 to chairlift1. It is difficult to explain why

DA-Faster is less accurate in one direction (chairlift2 → chairlift1) than in the other direction (chairlift1 → chairlift2). One assumption could be that in DA-Faster, the RPN is better trained on chairlift1 since in this case the background is less cluttered. Thus, when applying it on chairlift2, the adaptation process tends to promote features from the foreground and both the proposal and the classification are good. On the contrary, if the RPN is trained on chairlift2, it will rely on cluttered features that are removed with the global adaptation and thus, for DA-Faster, the proposals will be bad on chairlift1, leading to an important residual domain shift in the results. On the contrary, in our method, since the RPN is directly adapted, the residual shift is lower (see Figure 4.1).

Table 4.3 – Detection results on the chairlifts dataset. First, adaptation from chairlift1 to chairlift2, and second adaptation from chairlift2 to chairlift1.

Method	chairlift1 → chairlift2			chairlift2 → chairlift1		
	APcoco	AP50	AP75	APcoco	AP50	AP75
Faster R-CNN	30.34	99.49	0.30	36.56	98.98	9.86
DA Faster R-CNN	50.51	99.50	**33.4**	42.56	98.99	11.1
Ours	**50.93**	**99.99**	30.7	**48.83**	**99.00**	**45.6**

Fifteen-Six (FS)

This setting is intended to evaluate the ability of our method to detect dangerous situations in the practical context of the Mivao project. We use the *tiny chairlifts dataset* presented in Section 1.2.3 in the *FS* configuration presented in Section 3.2.2, and we evaluate the results in term of accuracy to be able to compare the score with a baseline classifier.

The training set is composed of images from 15 chairlifts training sets (source domain), and the test set is composed of the images of the remaining 6 chairlifts test sets (target domain). Note that there is only 300 labeled training images in the source domain and 180 unlabeled training images in the target domain (see Figure 4.5c).

In Chapter 3, we trained Faster R-CNN on the source training images and compared its performance to an image classifier with the same backbone when they are tested on the target test set. Here, we compare with the domain adaptation approaches trained with unlabeled images from the target domain in addition to labeled images from the source domain. In Table 4.4, we recall the classification accuracy of Faster R-CNN and baseline classifier, and we compare them to the adaptation methods. Our adaptation method enhanced Faster R-CNN accuracy from 87.12% to 90.20%. While DA-Faster accuracy dropped to 83.39%. Which confirmes that the RPN adaptation is crucial to adapt the detector in case of a domain shift between chairlifts.

Table 4.4 – Classification accuracy of Faster R-CNN, the baseline classifier, DA Faster and our adaptation method, with VGG16 backbone averaged over all the target chairlifts in the *tiny chairlifts dataset* using the configuration *FS*.

Approach	classification accuracy
Image classification	85.71
Faster R-CNN	87.12
DA Faster	83.39
Ours	90.20

4.5 Conclusion

In this chapter, we have presented our approach to solve the problem of domain adaptation for object detection. After an analysis of the complete workflow of the classical Faster R-CNN detector, we have proposed to adapt the features pulled from this network at two different levels: one adaptation at a global level in the Region Proposal Network and one adaptation at the local level for each bounding box returned by the RPN. We have shown that these two

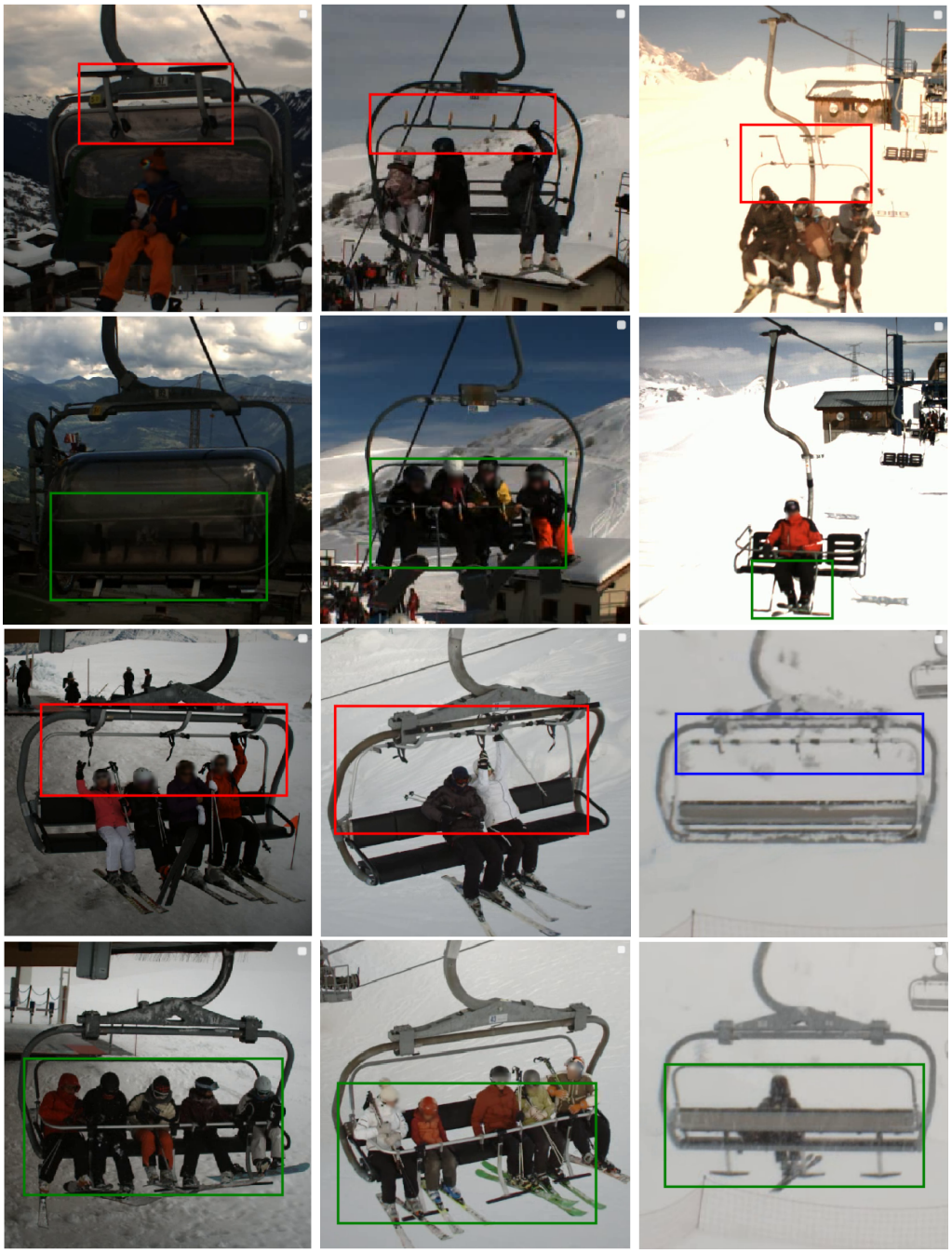


Figure 4.7 – Example images from target chairlifts, first and second rows from left to right C_0, C_1 and C_2, third and fourth rows from left to right C_6, C_9 and C_{12}. The box annotations (open:red and closed:green) are provided for illustration.

adaptations are complementary and provide very good detection results. We have tested our solution on two different applications, namely the autonomous driving and chairlift safety. For the latter, the results are particularly promising, where the generalization capacity has been significantly improved. Domain adaptation from one chairlift to another chairlift is not the best choice because it depends strongly on the similarity between the two chairlifts. However, adapting a model trained on a different set of chairlifts is likely to perform very well on the new target chairlifts. Only a few unlabeled images of the target domain are needed. As future works, we could test more accurate adaptation procedures such as the approaches presented in [99, 101]. These methods could help in the learning step to reach stable solutions, which is a strong weakness of the domain adaptive Faster R-CNN. Furthermore, it could be interesting to adapt the features at different depths of the network as recommended by [108].

Chapter 5

Mask-Guided Image Classification with Siamese Networks

In previous chapters, we have seen that a straightforward solution to the chairlift safety problem would be approaching it as an object detection problem. In this case, it is necessary to provide bounding box annotations that inform the network about the regions in the image containing the crucial elements. In this chapter, we propose an original solution that does not require any bounding box annotation. Our solution could be applied to CNN-based image classification tasks where the class of each image depends on small details it contains. The main idea is to inform the classifier of the elements in the image to focus on to make its decision. For this purpose, we resort to a Siamese architecture and feed the network with an image and a binary mask representing the important elements. We also study the performance of the models with a different number of annotated examples and different model depths. This work is presented in a paper [38] in the *International Conference on Computer Vision Theory and Applications, 2020.*

5.1 Introduction

In the context of Mivao project to ensure safe chairlifts boarding, considering that the safety bar has to be closed when the chairlift leaves the boarding station, our goal is to classify the images into images with open safety bar (called hereafter open images) and images with closed one (closed images). Thus, the class of an image is related to the position of a small number of pixels (the safety bar) that can be very hard to see in typical images and whose shape depends on the chairlift.

Since the safety bar is a non-deformable object which is always observed with the same viewpoint for a given chairlift, we can create two binary masks that represent its shape when it is open (open mask) and when it is closed (closed mask). Each time a new chairlift is installed, the operator can easily create these two mask images by acquiring one image of each class (open and closed) and by drawing two binary masks representing the shape of the safety bar (see Figure 5.1). These masks are already created to detect the safety bar in Sivao system provided by Bluecime. Figure 5.2 shows that the open (resp. closed) mask of a chairlift is not perfectly superimposed with all the open (resp. closed) images of this chairlift. But it gives a coarse idea about the shape of the safety bar and its relative location in the image.

The main point of our work is to find the best way to introduce this knowledge in the network. For this purpose, we propose to use Siamese networks and provide pairs of images as inputs; each pair consists of the colored image to be classified as well as a binary mask where important details of the image appeared in white over a black background. The Siamese architecture used allows controlling the features extracted from the colored image by forcing them to be similar to the features extracted from the binary mask. We found that this approach forces the network to concentrate on the pixels around the safety bar in the image to classify it. This is a way to decrease the difficulty of the classification task so that a small network with few parameters can solve the problem without requiring a lot of labeled data. To the best of our knowledge, this is the first approach to guide the network with a binary mask for a classification task.

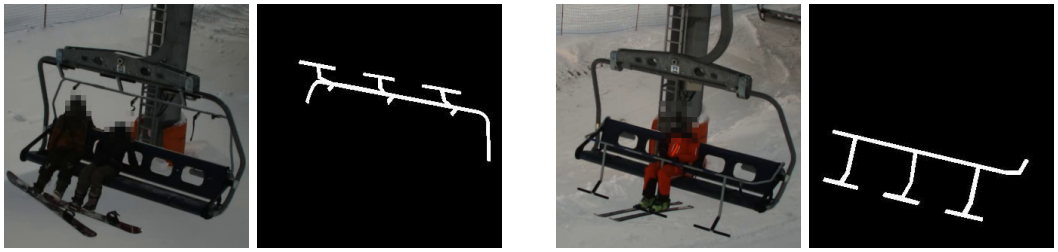
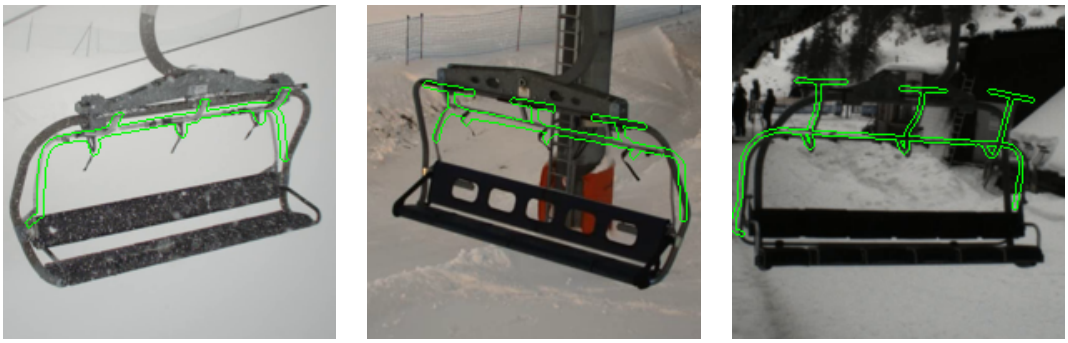


Figure 5.1 – Two images of the same chairlift C_{16} and the corresponding masks. Left: the safety bar is open, right: the safety bar is closed.

A second advantage of using a specific binary mask for each chairlift is that the Siamese network is not trying to learn general features that should work on all the chairlifts, but instead it learns specific features adapted to each chairlift (each mask). A good way to get more accurate results for each chairlift is to focus on the specificity of each chairlift and not on the invariance of features across chairlifts.

The open mask of chairlift C_2 superposition with 3 different open images of C_2.



Open masks superposition with open images from chairlifts C_9, C_{16} and C_{19}.

Closed masks superposition with closed images from chairlifts C_9, C_{16} and C_{19}.

Figure 5.2 – Superposition of masks and the corresponding images.

5.2 Related work

The most similar approach to ours deals with a person re-identification task [131], where the idea is to help the network extract features only from the body of the person in the image and not from the cluttered background. In this aim, the authors propose to use a binary mask of the person to create three images: the full image, the body image, and the background image. Then a triplet loss is used to bring the features of the full image closer to those of the body alone and to move away the features of full images from those of the background image. Thus, the network is trained to automatically extract the most important features (*i.e.* from the body only) from the full image. This approach requires designing a triplet loss to extract features from the body and a Siamese network to bring closer images from the same person and move images of different people apart. This complex architecture is not adapted to our problem with few labeled images.

Conditional networks could be an inspiring solution to exploit the available data [151, 135, 136]. For example, Zhou *et al.* propose to introduce geometric constraints in the output of their network designed to estimate the 3D human pose from non-calibrated 2D images [135]. Since the problem is tough to solve, the authors add constraints on the relative size of the human bones such as upper and lower arms have a fixed length ratio, left and right shoulder bones share the same length, *etc.* Zhao and Snoek propose to modulate the RGB features of a video with optical flow features to improve the action detection accuracy [136]. The proposed motion condition and motion modulation layers incorporate motion and modulate the contribution of the RGB features. Such conditional networks require the different features (optical flow and RGB) to be well spatially registered, which is not the case for our images and binary masks.

Another way to provide additional information to a network is to add branches that attempt to solve certain auxiliary tasks while the main branch concentrates on the main task. If the auxiliary tasks are well-chosen, they will help to solve the main task in such a multi-task network. For example, Lee *et al.* try to predict some other information, in addition to the main classical detection task (prediction of the location and class of the objects), such as the area portions occupied by each ground truth box within a window, the distances from the center of the box to those of other boxes or a binary mask between foreground and background [122]. All these data are available from the ground truth labels but trying to predict them helps in solving the main detection task. Likewise, Channupati *et al.* improve the results of their semantic segmentation network by adding a branch that estimates the depth of the pixels as an auxiliary task [123]. Since the depth was available in their used dataset, they propose to exploit it at training time and create a multi-task network. They just remove the depth estimation branch at test time and notice that the main task (semantic segmentation) is improved. These last solutions are specific to the considered tasks and available data at training time. They can not be applied to our problem.

Part of related work concerns the use of Siamese networks for comparing multimodal images. Indeed, by providing pairs of images as input and designing specific losses, the Siamese networks are a smart solution to compare patches from different modalities (color, infra-red, thermal, sketch, *etc.*). When the two sub-networks share their weights, the idea is to extract features that are common to the two modalities, while when the two sub-networks are different (Pseudo-Siamese network) the aim is to discover the features specific to each modality. En *et al.* propose to exploit the benefits of these two approaches in a single three-stream network [141].

Siamese networks are also widely used in the context of object tracking [138]. The idea there is to learn invariance of object representation (as in [152, 153]) across time. By providing pairs of images representing the same object with different viewpoints, scales, orientations, or light source conditions, the network is trained to extract features that remain stable across all these transformations. Our goal is slightly different since we are using Siamese networks to help the model to concentrate on some parts of the images while extracting features.

5.3 Mask-guided image classification approach

The principle of our approach is to use a Siamese network structure [154] to learn a function $F(X)$ mapping an input image X to a low dimensional feature space well suited to compare this image with two binary masks corresponding to the specific classes to be tested. Once trained, the

Euclidean distance in the feature space can be used to decide whether the input image belongs to the first or the second class (see Figure 5.3).

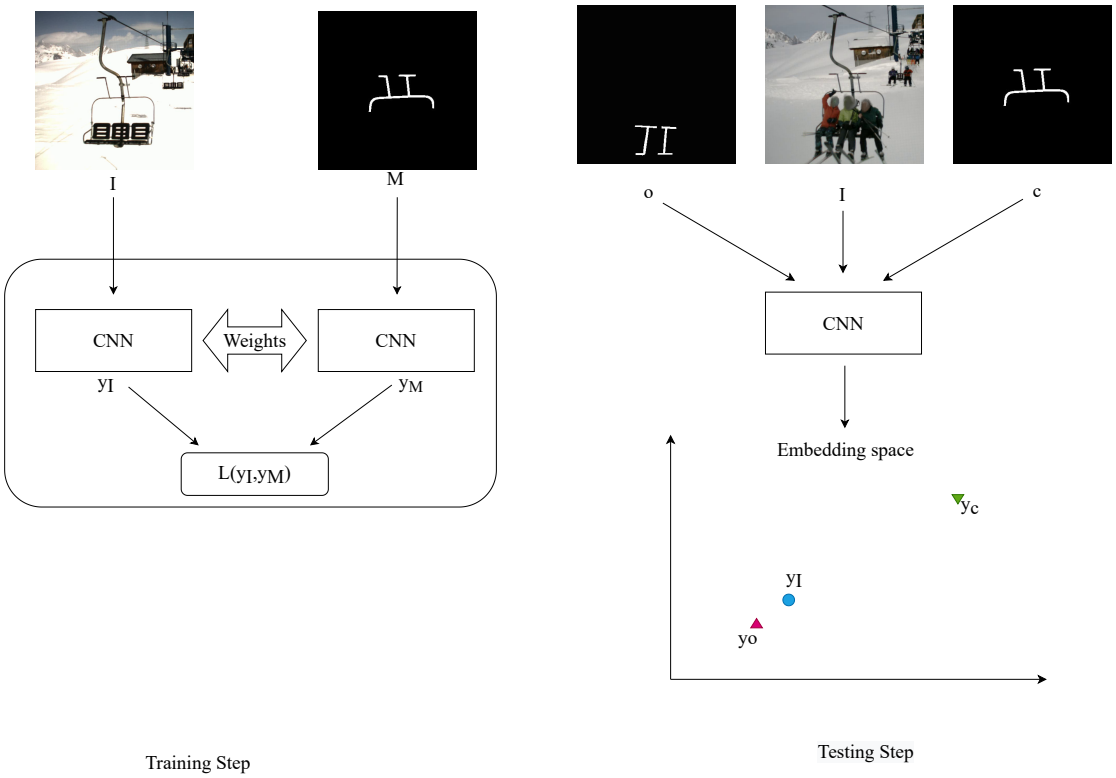


Figure 5.3 – Principle of our Mask-Guided image classification approach using Siamese networks.

More precisely, in our problem, we denote the colored image I, and the binary mask M which could be either open o or closed c. The Siamese structure comprises two sisters CNNs of the same architecture sharing their weights. Each of the two inputs X is transformed into a low dimensional feature vector $F(X)$ through the CNN.

At training time, the first input is a colored image I belonging to one of the two classes, and the second one is a binary mask M. The two outputs $y_I = F(I)$ and $y_M = F(M)$ are compared through a contrastive loss function $\mathcal{L}$ defined by [155]:

$$\mathcal{L}(y_I, y_M) = \alpha \|y_M - y_I\|^2 + (1 - \alpha) \max(1 - \|y_M - y_I\|, 0)^2 \tag{5.1}$$

where $\|.\|$ denotes the L_2 norm, $\alpha = 1$ if the class of the image is the same as the class of the mask and $\alpha = 0$ otherwise.

At test time, only one branch of the network is used to compare the distance of a test image I from both open and closed masks o and c, respectively, in the feature space. The inferred image class $\hat{Y}$ is then this of the nearest mask. Formally:

$$\hat{Y} = \arg\min_{M \in \{o,c\}} \|y_M - y_I\|^2 \tag{5.2}$$

where, $y_I = F(I)$ and $y_M = F(M)$ are the output vectors of the image and the mask, respectively.

In our application context, we address a more general situation where images and masks belong to different domains. Specifically, in the chairlift safety scenario, we want to process with the same model, images coming form N different chairlifts of the ski resort (or even from different ski resorts). Thus, each set of images extracted from a specific chairlift $C_i, i = 1, \ldots N$ concerns vehicles of a different shape, different number of seats and was taken from a different viewpoint (see Figure 1.8). We suppose that for each chairlift C_i, the two binary masks o_i and c_i respectively associated with the open and closed safety bar are available.

Then, the training and testing approaches proposed above can be generalized to the multi-domain situation. At training time, image-mask pairs from all the domains are given to the Siamese network, ensuring that the image and the mask belong to the same chairlift. Like in

the single domain situation, a pair is positive if the image and mask labels are of the same class and negative otherwise. The learned CNN function $F(X)$ allows projecting images and masks of all chairlifts in the same embedding space. Given the i^{th} chairlift, the training pairs are denoted $\{I_i, M_i\}$ where I_i is an image and M_i is a mask, and the two outputs $y_{I_i} = F(I_i)$ and $y_{M_i} = F(M_i)$ are compared through the contrastive loss:

$$\mathcal{L}(y_{I_i}, y_{M_i}) = \alpha \|y_{M_i} - y_{I_i}\|^2 + (1-\alpha)\max(1 - \|y_{M_i} - y_{I_i}\|, 0)^2 \tag{5.3}$$

At test time, each image I_i of chairlift C_i is compared to the two masks of its corresponding chairlift in the feature space, to infer its class:

$$\hat{Y} = \arg\min_{M_i \in \{o_i, c_i\}} \|y_{M_i} - y_{I_i}\|^2 \tag{5.4}$$

where, $y_{I_i} = F(I_i)$ and $y_{M_i} = F(M_i)$ are the output vectors of the image and the mask, respectively.

5.4 Experiments

To evaluate the performance of our approach, we have explored a range of simple to challenging configuration options in terms of network depth and the amount of labeled data, in the context of Mivao project.

5.4.1 Experimental setup

Chairlifts datasets. The dataset used in the following experiments is Mivao 2018 described in Section 1.2.3 composed of 21 different chairlifts with different shapes, sizes, points of view, backgrounds, and weather conditions. The images of each chairlift are separated into training and test sets. The model parameters are learned by using the training images along with the corresponding masks from all chairlifts together (this is equivalent to the configuration *All* explained in Section 3.2.2). Similarly, we use the test images to check the accuracy of the model after training.

Experiments were performed for a variety of difficulty settings, and this is why we need different versions of the dataset to test each setting. Therefore, we use the following datasets with different amounts of labeled data, all these datasets are already presented in Chapter 1, see Section 1.2.3 for more details. Table 1.5, shows the distribution of images between the two classes in each dataset.

- ***Large chairlifts dataset.*** The two classes are not perfectly balanced in the dataset in general, and the difference between the number of images in each class is more important in some chairlifts like C_3, C_7 and C_{19}. The total number of training images is 51774, and the total number of test images is 14252.
- ***Medium chairlifts dataset.*** For each chairlift, we have randomly chosen 500 images from the training sets of the large dataset, the total number of training images is 10500. The test set remained the same for a fair comparison. Since the smaller sets of 500 images are randomly selected from the previous training sets, the imbalance in the dataset still exists.
- ***Small chairlifts dataset.*** We have randomly chosen 100 training images from the training sets of the large dataset, the total number of training images is 2100. The test set is kept the same.
- ***Tiny chairlifts dataset.*** In this dataset, there are only 20 training images per chairlift, the total number of training images is 420. The test set is kept the same. Because the size of the dataset is too small, and there is an imbalance in the classes in the original dataset we ensured that there are 10 images per class for each chairlift. The test set remained the same.

Baseline network. The Siamese model has two identical networks that share weights. We propose to use the following two architectures:

1. A shallow architecture composed of:

   - A convolutional layer with 32 filters of size 3×3 and ReLU activation,
   - A convolutional layer with 64 filters of size 3×3 and ReLU activation,
   - A MaxPooling layer with a window of size 2×2,
   - A fully connected layer with 2 outputs.

   Since the embedding space has only two dimensions, this exact same architecture is used as a baseline classifier by adding a Softmax activation. Considering the weights are shared between the two sister networks in our Siamese model, the number of parameters to learn in both models (baseline classifier and our Siamese model) are the same and equal to 1.2 million. All layers are randomly initialized.

2. A deep architecture composed of:

   - VGG16 convolutional part (13 layers),
   - A fully connected layer with 4096 outputs,
   - A fully connected layer with 1024 outputs.

   The baseline classifier has the same architecture, augmented with a two neurons classification layer with a Softmax activation. The convolutional part of VGG16 [153] pre-trained on IMAGENET [142] and the additional fully connected layers are randomly initialized. The number of parameters of the Siamese model is nearly the same as the one of the baseline classifier (the latter having 2048 more weights).

All models are trained using back-propagation and the stochastic gradient descent algorithm. The number of epochs is set to 1000, the learning rate to 10^{-5}, the learning rate decay to 10^{-8}, and the momentum to 0.9. While the inputs of the baseline classifier are colored images, our Siamese model takes as input pairs of: (i) colored image of one chairlift and (ii) one of the two corresponding masks from the same chairlift. During training, we make sure that the positive and negative pairs are well balanced, so that we consider:

- 50% positive pairs: (open image - open mask) and (closed image - closed mask);
- 50% negative pairs: (open image - closed mask) and (closed image - open mask).

Since the Siamese model is perfectly symmetric with shared weights between the two sisters networks, the images and masks must have the same size. Consequently, we have transformed the masks to have their depth equal to 3 (as RGB images) by concatenating it three times along the channel dimension.

5.4.2 Shallow architectures experimental results

The purpose of these experiments is to prove the validity of our approach using a shallow randomly-initialized network. Moreover, to show that at test time, it can automatically adapt itself to the concerned chairlift. Furthermore, we investigate the effect of the number of the training images on the results using the different datasets explained in the previous section to train the models. Table 5.1 shows the obtained accuracy. Figure 5.4 illustrates the average accuracy of the shallow models achieved with different datasets.

We notice that our Siamese model outperforms the corresponding baseline classifier for almost all the chairlifts and provides an average accuracy of 99.48% over the whole dataset, compared to 98.32% obtained by the corresponding baseline classifier, when they are trained on the *large chairlifts dataset*. Since these two models have the same architecture and number of parameters, these results clearly show that inserting the location of the safety bar with a binary mask into the model helps to extract more accurate features. However, training the Siamese model is longer as we input each image twice (with the two masks); this can be overcome by randomly selecting a subset of the pairs instead of taking all of them.

When the Siamese model is trained on the *medium chairlifts dataset*, which is five times smaller than the *large chairlifts dataset*, the accuracy decreased only by 1.31 percentage points. In comparison, the accuracy of the baseline classifier decreased by 3.32 percentage points using

the smaller dataset, indicating that the Siamese model is less sensitive to changes in dataset size at this scale.

Our Siamese model outperforms the corresponding baseline classifier when the models are trained on the *small chairlifts dataset*. Compared to the Siamese model trained on the *large chairlifts dataset* and the one trained on the *medium chairlifts dataset*, the accuracy decreased by 6.14 and 4.83 percentage points, respectively, while the accuracy of the baseline classifier decreased by 7.63 and 4.31 percentage points respectively. Here, the Siamese model and the baseline classifier seem to be equally sensitive to changes in the size of the dataset at this scale.

The most challenging case is when the models are trained on the 420 training images of the 21 chairlifts of the *tiny chairlifts dataset*. Compared to the Siamese model trained on the larger datsets, the accuracy decreased to 81.05, while the accuracy of the baseline classifier decreased to 80.45, and therefore the Siamese model becomes more sensitive as the dataset gets smaller, but it still outperforms the baseline classifier by a small margin.

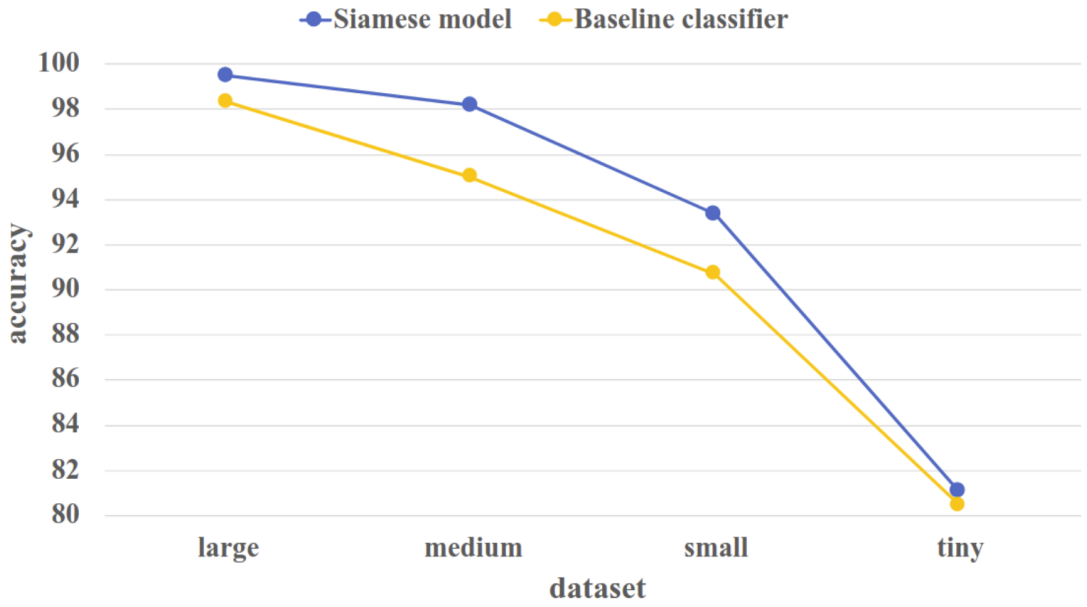


Figure 5.4 – The accuracy (per training dataset) of the Siamese model and the corresponding baseline classifier with shallow architectures.

Observing the embedding space

Since we have chosen a small embedding space with two dimensions, we can project each image and mask in this space and observe the distributions. Figure 5.5 shows the distributions of test images using the 4 shallow models (*i.e.* based on the dataset size it is trained on) for 5 different chairlifts with the corresponding masks. In this figure, we can see the impact of the contrastive loss on the distributions. Indeed, this loss brings closer the open (resp. closed) images around the corresponding open (resp. closed) mask and move them away from the closed (resp. open) images and closed (resp. open) mask. This is worth mentioning that there is a single 2D embedding space for each model, and that all these points could have been drawn in a single plot per model, but for the sake of clarity, we have preferred to display one plot per model for each chairlift. We remark that the larger the dataset the model is trained on, the better the separation between the two classes and the higher the accuracy.

The distributions of the masks of the 21 chairlifts are shown in Figure 5.6. This mask distributions shows two important things. First, the two masks of each chairlift are far away from each other. This is due to the contrastive loss that keeps open images and masks away from closed images and masks. Second, although there is no constraint in the loss forcing the open masks (resp. closed masks) to be near each together, we notice that this is almost the case and we can see two clouds, one with the open masks and one with the closed masks.

The results of the previous experiments clearly show that it is very interesting to guide the network with a binary mask to highlight the most important part of the images. In the next

Table 5.1 – Accuracy of the Siamese model and the corresponding baseline classifier with shallow architectures, trained on the 21 chairlifts from the *large, medium, small* and *tiny chairlifts dataset* and then test on each chairlift independently.

Chairlift	*large*		*medium*		*small*		*tiny*	
	baseline classifier	Siamese model	baseline classifier	Siamese model	baseline classifier	Siamese model	baseline classifier	Siamese model
C_0	98.24	**98.99**	95.72	**97.98**	93.95	**95.59**	88.92	**93.20**
C_1	99.44	**99.67**	97.00	**98.00**	93.22	**95.06**	89.22	**90.11**
C_2	96.79	**98.61**	93.43	**97.45**	90.36	**91.17**	81.31	**85.04**
C_3	98.02	**99.40**	96.03	**98.41**	88.29	**91.07**	64.88	**77.38**
C_4	96.32	**99.33**	94.56	**96.82**	91.30	**91.47**	**87.96**	81.44
C_5	98.77	**99.47**	96.32	**98.68**	91.75	**94.04**	**77.02**	75.61
C_6	98.80	**99.49**	97.09	**98.72**	95.04	**96.24**	91.45	**91.97**
C_7	97.66	**99.51**	93.46	**97.16**	**94.94**	93.65	**84.83**	52.77
C_8	96.64	**99.51**	86.89	**96.26**	80.50	**86.46**	68.26	**76.92**
C_9	98.89	**99.79**	95.96	**97.35**	92.62	**95.26**	94.01	**95.26**
C_{10}	98.92	**99.58**	96.59	**98.63**	91.69	**94.35**	**76.97**	75.56
C_{11}	97.68	**99.23**	97.30	**99.03**	**96.14**	95.75	**96.91**	92.08
C_{12}	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**	99.20
C_{13}	98.70	**100.0**	97.66	**98.70**	95.84	**96.23**	81.82	**92.34**
C_{14}	99.75	**100.0**	95.18	**99.20**	91.72	**96.63**	**86.16**	81.83
C_{15}	99.57	**99.96**	93.53	**99.20**	77.14	**87.42**	52.55	**63.77**
C_{16}	96.95	**98.91**	89.39	**96.58**	79.80	**90.92**	72.82	**73.26**
C_{17}	97.77	**99.26**	96.28	96.28	94.80	**95.91**	91.08	**92.38**
C_{18}	98.85	**99.42**	97.69	**99.13**	90.96	**96.63**	62.12	**71.35**
C_{19}	**99.80**	99.69	98.36	**99.49**	**96.93**	93.46	**78.53**	72.70
C_{20}	97.11	**99.19**	87.53	**98.88**	77.47	**82.92**	62.71	**67.80**
Avg.	98.32	**99.48**	95.00	**98.17**	90.69	**93.34**	80.45	**81.05**

experiment, we will test deep models on the *tiny chairlifts dataset* to check if the pre-trained deep networks can provide better results than those provided by the randomly-initialized shallow networks when few labeled data are available.

5.4.3 Deep architectures experimental results

In this experiment, we attempt to improve the results on the *tiny chairlifts dataset* by using a deep network instead of the shallow randomly-initialized one, and compare both models (shallow and deep). Testing the deep network on our Siamese model is a good way to check if the shallow randomly-initialized network can provide good results as a deeper and pre-trained network despite its architecture that is not at all optimized. We also compare the Siamese model with its corresponding baseline classifier which has the same architecture. The results of the two deep models are presented in Table 5.2. We notice the accuracy of our Siamese model is improved from 81.05% to 98.31% using the deep network, which is expected due to the fact that the shallow architecture has only 2D embedding space. However, our deep Siamese model is equivalent to the corresponding baseline classifier which is also improved from 80.45% to 98.14%. The baseline classifier has improved more, probably because VGG16 architecture was initially designed and pre-trained for classification.

5.5 Conclusion

In this chapter, we have presented an original solution to introduce additional data in a network. Considering a classification problem where the class of each image depends on the location of a thin bar, we have proposed to represent the knowledge of the shape and coarse position of this bar with a binary mask. This mask and the colored image are the two inputs of a Siamese network that extracts and projects their features in an embedding space. We have applied this solution to the chairlift safety problem, where the images have to be classified whether they have

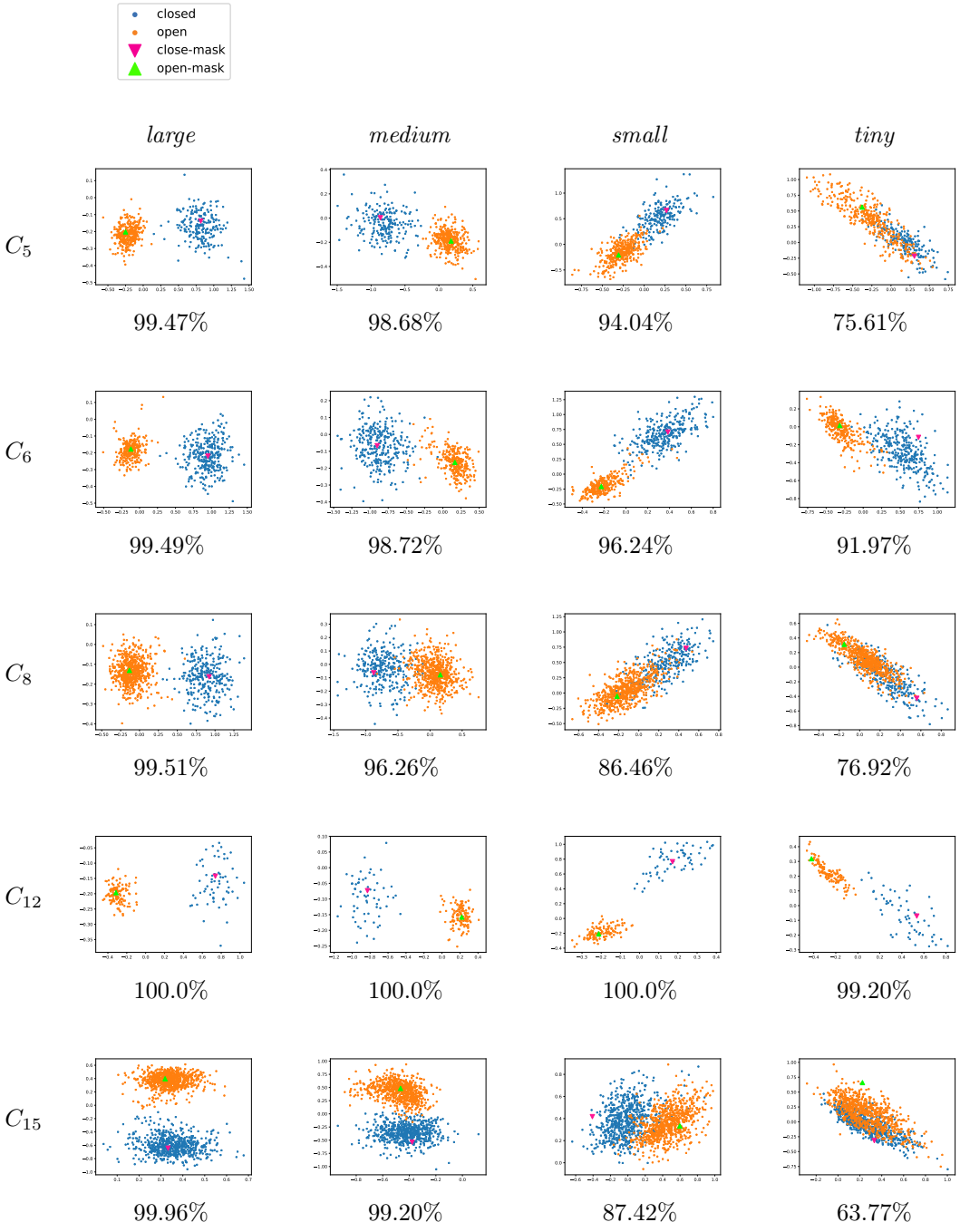
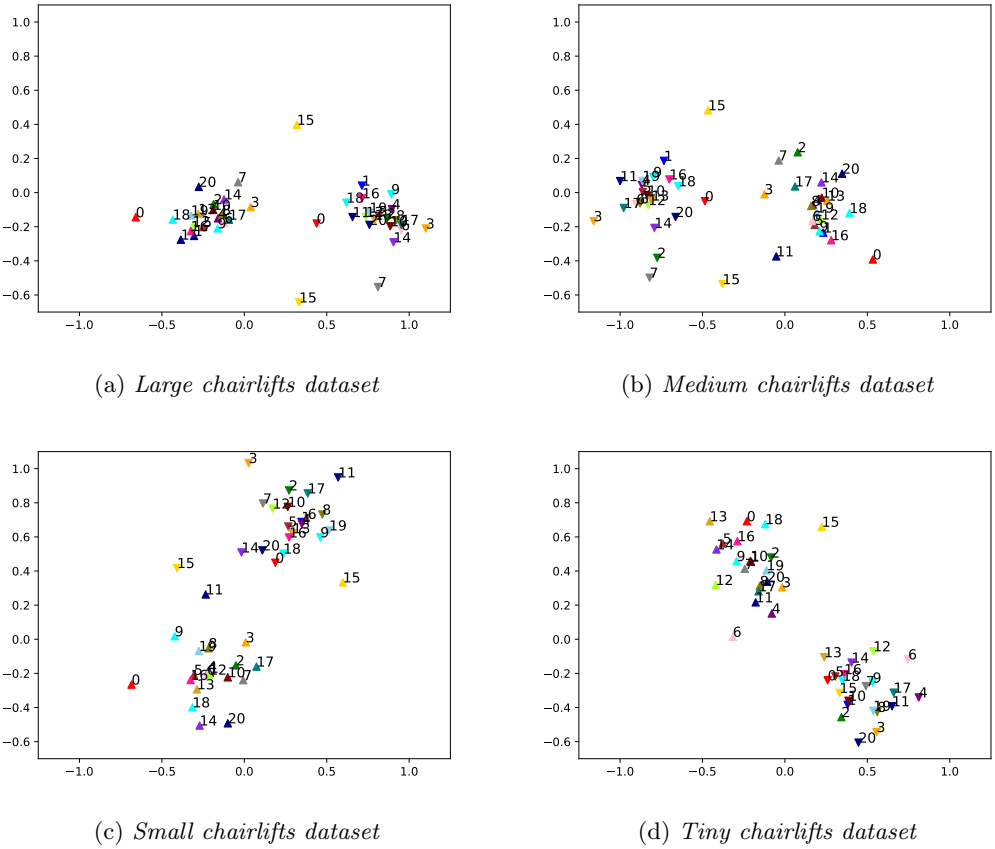


Figure 5.5 – Projections in the 2D embedding space of the masks and test images of 5 different chairlifts C_5,C_6,C_8,C_{12} and C_{15}, after training a Siamese model on the *large, medium, small* and *tiny chairlifts datasets.* We provide the accuracy under each sub-figure.

a safety bar open or closed. The training step consists in extracting features from closed images that are similar to features of the closed mask, but different from the features of the open mask (and the reverse for the features extracted from open images). During the test step, we just extract features from each image and check if they are nearer to the features of the open or of the closed masks. Experimental results showed that this architecture is able to extract specific features from each chairlift. Indeed, a single Siamese network trained on 21 different chairlifts provided very good results on each of these chairlifts.

Furthermore, when the training set is large enough, our shallow Siamese network provided as good results as much deeper networks such as VGG16 trained on a tiny dataset. In Mivao,



(a) *Large chairlifts dataset*

(b) *Medium chairlifts dataset*

(c) *Small chairlifts dataset*

(d) *Tiny chairlifts dataset*

Figure 5.6 – Distribution of all the masks of the 21 chairlifts in the embedding space, after training a Siamese network on the *large, medium, small* and *tiny chairlifts datasets*. Up triangles are for open masks while down triangles are for closed masks. The legend numbers are referred to the number of each chairlift.

if a large dataset is available and labeled with image-level annotations, a Siamese model with shallow architecture will give satisfactory results and a deep network that requires a lot of time to be trained is not necessary. However, when the labeled dataset is small, a drop in accuracy will occur if a shallow architecture is used and a deep model is then a necessity. In the next chapter, we will assess the generalization ability of our approach by testing our Siamese network on new unseen chairlifts with different 3D geometries.

Table 5.2 – Accuracy of the Siamese model and the corresponding baseline classifier with deep architectures, trained on the 21 chairlifts from the *tiny chairlifts dataset* and then tested on each chairlift independently.

chairlift	baseline classifier	our Siamese model
C_0	**98.11**	96.47
C_1	99.56	**99.67**
C_2	**98.69**	97.96
C_3	**89.38**	88.89
C_4	98.66	**99.67**
C_5	99.04	**99.82**
C_6	99.66	**100.0**
C_7	98.89	**99.14**
C_8	96.05	**96.21**
C_9	99.79	**100.0**
C_{10}	**99.71**	99.17
C_{11}	99.61	**100.0**
C_{12}	100.0	100.0
C_{13}	100.0	100.0
C_{14}	98.21	**98.39**
C_{15}	91.19	**98.20**
C_{16}	**98.62**	97.82
C_{17}	**99.44**	97.77
C_{18}	99.23	**99.62**
C_{19}	**98.36**	97.34
C_{20}	**98.73**	98.37
Avg.	98.14	**98.31**

Chapter 6

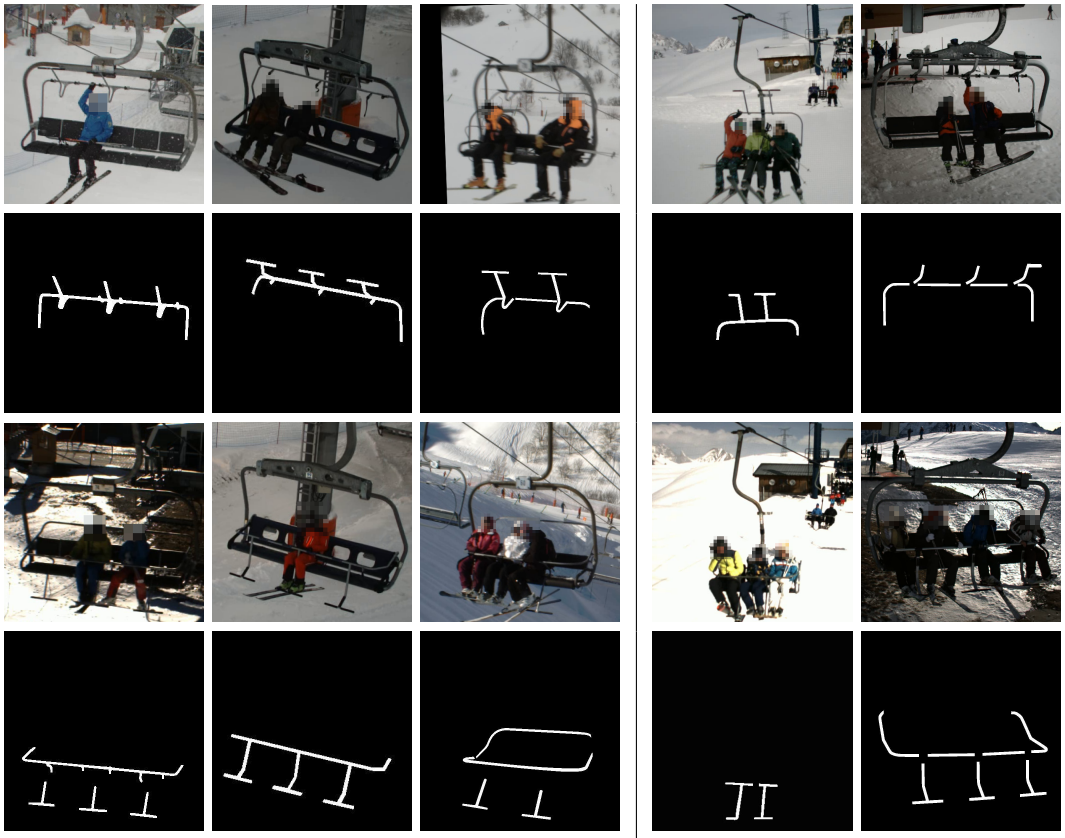
Domain Generalization with Geometric Constraints

In this chapter, we propose a solution to improve the learning process of a classification network when less labeled images are required. In Chapter 5, we have shown that a Siamese architecture is a good solution to exploit the geometrical constraints provided by binary masks. In this chapter, we show that this original learning process provides the model a significant generalization power. This approach allows to challenge our model by training and testing it on different domains (source and target) and showing promising results. Furthermore, we propose a fine-tuning step using only two binary masks from the target domain.

6.1 Introduction

The application context of our work is Mivao project to secure the boarding on chairlifts in ski resorts, the project is presented previously in Section 1.1. In this scenario, we have different chairlifts spread over different ski resorts, each chairlift corresponds to a specific domain. The chairlifts for which images have been annotated are the source domains and the chairlifts for which no images have been annotated are the target domains. In addition to images and annotations, each chairlift comes with information about the geometry of the safety bar. This information is given by two binary masks representing the shape of the safety bar when it is open or closed (see Figure 6.1).

Giving image-mask pairs, a Siamese model learns a specific embedding space where the features of images with open safety bar are similar to those of open mask, and features of images with closed safety bar are similar to those of closed mask. As shown in Chapter 5, feeding the model with binary masks is a smart way to inform the network of the important elements it should focus on to make the final decision, thus helping the learning process. In this chapter, we assess and improve the generalization property of such a Siamese model. First, we show that a deep network can be used as a backbone in our Siamese architecture and boosts the results on the source data by using a small training set of only 300 images and 30 masks. Indeed, providing pairs of image-mask allows the model to learn one discriminator for each chairlift while projecting all the output features in the same embedding space allowing the different discriminators to collaborate. Second, we challenge the Siamese model by testing it on new, unseen chairlifts (target data) in two different settings, depending on whether the binary masks of the new chairlifts are available or not. If the binary masks of the new target chairlifts are not available, we propose to use virtual masks around which the source images concentrate, one virtual mask per class. Then use these masks to classify the target images. If the binary masks of the new target chairlifts are available, we propose to fully exploit the information they contain to classify the new images. Furthermore, we propose a fine-tuning solution to boost the generalization power of the model using only binary masks from the target domain. It is worth mentioning that, unlike the domain adaptation approaches, none of these solutions are using the target feature distributions to improve the results. The primary goal of this work is to show that geometric constraints can boost the generalization property when they are adequately exploited.



Some source images and their corresponding masks

Some target images and their corresponding masks

Figure 6.1 – Some source and target images and their corresponding masks. Each column contains images and masks from the same chairlift (top:open and bottom:closed).

6.2 Related work

The goal of domain adaptation is to decrease the discrepancy between the source and target distributions to optimize the performance of the model on target data. Usually, domain adversarial training is used to move closer the distributions [99, 101].

Nevertheless, when the target data is not available while learning the model, people resort to zero-shot domain adaptation [156, 117, 118, 114]. Yang and Hospedales propose associating each domain (source and target) with a vector of discrete parameters called a semantic descriptor [117]. Then, they use a two-branche network whose inputs are the features of the sample and the domain descriptor from this sample. The output of the network (the predicted sample class) is a fusion of the outputs of the two branches. This is a way to adapt the classifier to the domain provided as input. For a new unseen target domain, given its semantic descriptor, the network is able to predict the class of its samples accurately. This approach requires that the user can describe all the domains with a vector of discrete parameters. Kumagai and Iwata use a similar architecture but instead of using an attribute vector for each domain they propose to extract a latent domain vector from the set of features of each domain [118]. This approach requires the knowledge of the whole target features to start predicting the classes of the target data. Unlike the previous solutions that use domain features to predict labels, Shankar *et al.* propose to learn a domain invariant classifier [114]. Their original solution consists in artificially transferring data between domains to learn invariant features and they show that this augmented data helps to better generalize to unseen domains. In our case, the problem is slightly different from all the previous approaches since we have additional geometric features that are domain-dependent and that can be inserted in the model.

6.3 Domain generalization with geometric constraints

In Chapter 5, to insert geometrical constraints in our model, we proposed to resort to a Siamese model that learns a mapping that projects the images and masks into an embedding feature space, each mask corresponding to the specific classes to be tested (open or closed) [154]. At test time, Euclidean distances are evaluated in the feature space to decide whether the input image belongs to the first or the second class.

As illustrated in Figure 6.2, in our binary classification problem, we have a set of training images and two binary masks for each chairlift. For clarity, we illustrate the process in this figure with two different chairlifts but our model is actually trained with a set of different chairlifts. We denote o_{ij}, the j^{th} open image from the i^{th} chairlift and c_{ik}, the k^{th} closed image from the i^{th} chairlift. The two masks of the i^{th} chairlift are denoted o_i and c_i.

6.3.1 Recall our Siamese model principal

Training the model

At training time, a Siamese model, composed of two sister CNNs sharing their weights is used. As explained in Chapter 5, given the i^{th} chairlift, training pairs $\{I_i, M_i\}$ feed the Siamese model, each pair being composed of an image I_i and a mask M_i of the given chairlift. Each sister CNN is trained so that the two inputs are transformed into two vectors that will be similar if they are from the same class and different if they are from two different classes. Denoting F the function mapping the input image (or mask) to the corresponding output vector, the two outputs $y_{I_i} = F(I_i)$ and $y_{M_i} = F(M_i)$ are compared through a contrastive loss function $\mathcal{L}$ defined by [155]:

$$\mathcal{L}(y_{I_i}, y_{M_i}) = \alpha \|y_{M_i} - y_{I_i}\|^2 + (1-\alpha)\max(1 - \|y_{M_i} - y_{I_i}\|, 0)^2 \tag{6.1}$$

where $\|.\|$ denotes the L_2 norm, $\alpha = 1$ if the class of the image is the same as the class of the mask and $\alpha = 0$ otherwise.

By learning this model on multiple chairlifts and projecting all the source images and masks in the same embedding space, the model learns to automatically extract discriminant features specific to the safety bar in the images.

Testing on source domain

In this section, we consider the test data from the source domain, for which the binary masks representing the safety bar shapes are available, this is the case described in Chapter 5. Considering one test image of the i^{th} chairlift, we can use one branch of the learned Siamese model and feed it with the two masks of this chairlift o_i and c_i as well as the test image I_i, giving the respective outputs $y_{o_i} = F(o_i)$, $y_{c_i} = F(c_i)$ and $y_{I_i} = F(I_i)$ (see Figure 6.3). The inferred image class $\hat{Y}$ is deduced by considering the nearest mask, in terms of Euclidean distance, in the embedding space:

$$\hat{Y} = \arg\min_{M_i \in \{o_i, c_i\}} \|y_{M_i} - y_{I_i}\|^2. \tag{6.2}$$

6.3.2 Generalizations scenarios

For inference on the target domain, we consider two different cases:

- the corresponding target masks are not available;
- the corresponding target masks are available.

It is worth mentioning that in both cases the distribution of the target data is not available, and can not be used unlike classical domain adaptation. We could refer to these cases as a zero-shot adaptation process since no target images are used.

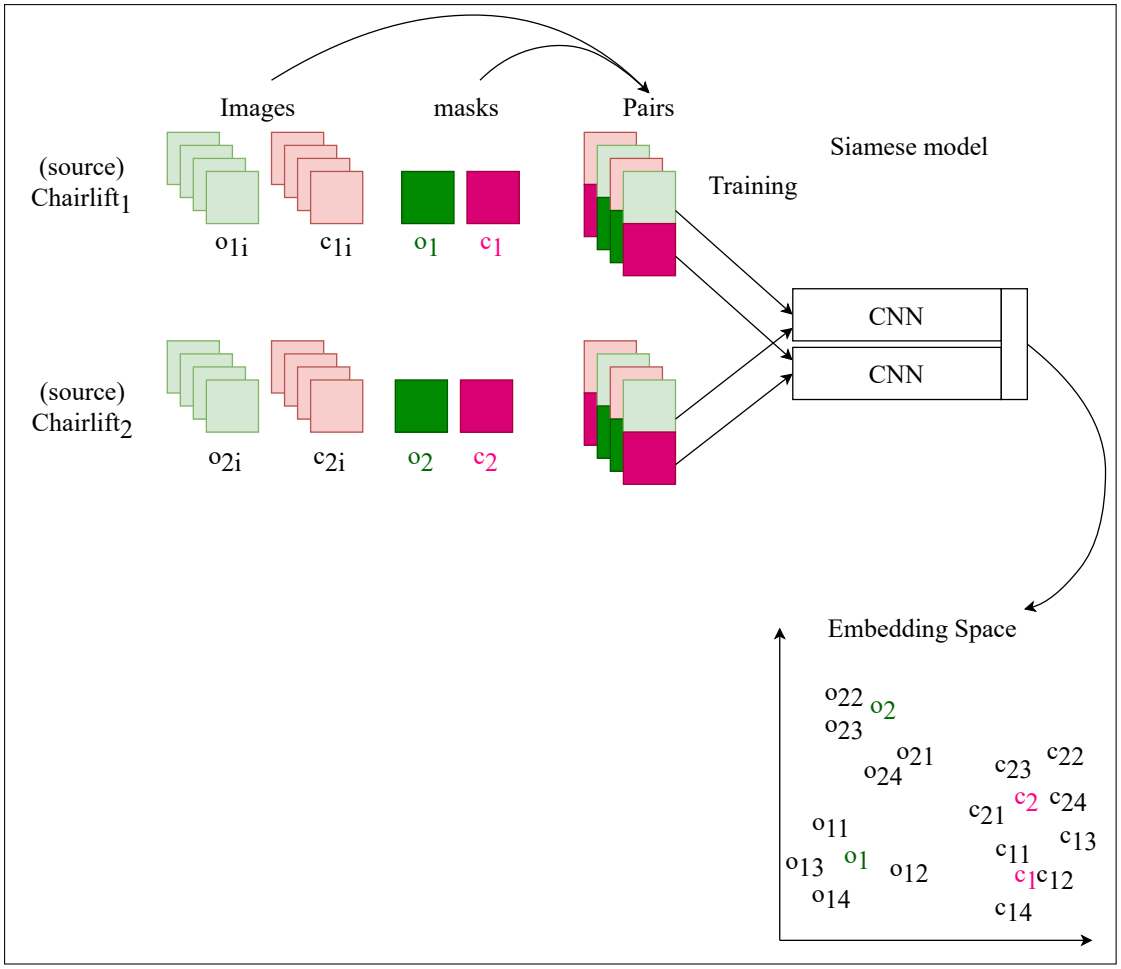


Figure 6.2 – Train on the source domain with image-mask pairs.

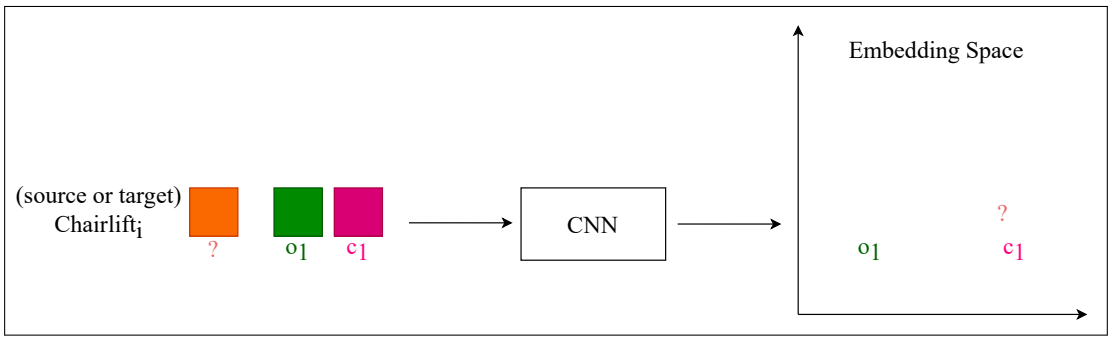


Figure 6.3 – Test on source or target domain, when masks are available.

Testing on target domain without target masks

In this section, we consider the most challenging case, where we would like to infer the class of images from a target domain chairlift, *i.e.* a chairlift that has not been used to train the model. Furthermore, we assume that we do not have any information about this chairlift, *i.e.* neither the image distribution in the embedding space nor the two binary masks informing about the safety bar shape and location.

By looking at the feature vectors of the source open and closed masks in the embedding space (see Figure 6.7), we notice that they are distributed in two well-separated clouds. Even if this was not a constraint of our training process, the network automatically discovered that the open (resp. closed) masks share geometric properties that help to discriminate them from the closed (resp. open) masks. This property is interesting since we can get a coarse idea about the

location of the feature vectors of any new unseen masks. To exploit this knowledge, we propose to use the average location of the source masks as an approximate location of the masks of a target chairlift. After reaching the convergence, we evaluate the average location of the open masks $o_a = \frac{1}{N}\sum_i^N F(o_i)$, where N is the number of source chairlifts, and the average location of the closed masks $c_a = \frac{1}{N}\sum_i^N F(c_i)$ in the embedding space. Thus, given an image I_k from the k^{th} target chairlift, without having its masks, we evaluate the distance between its feature vector and the two average feature vectors of the open and closed source masks. The nearest average mask provides us the class of this test image (see Figure 6.4):

$$\hat{Y} = \arg\min_{M_a \in \{o_a, c_a\}} \|y_{M_a} - y_{I_k}\|^2. \tag{6.3}$$

Where, $y_{I_k} = F(I_k)$ and $y_{M_a} = F(M_a)$ are the output vectors of the image and the mask, respectively.

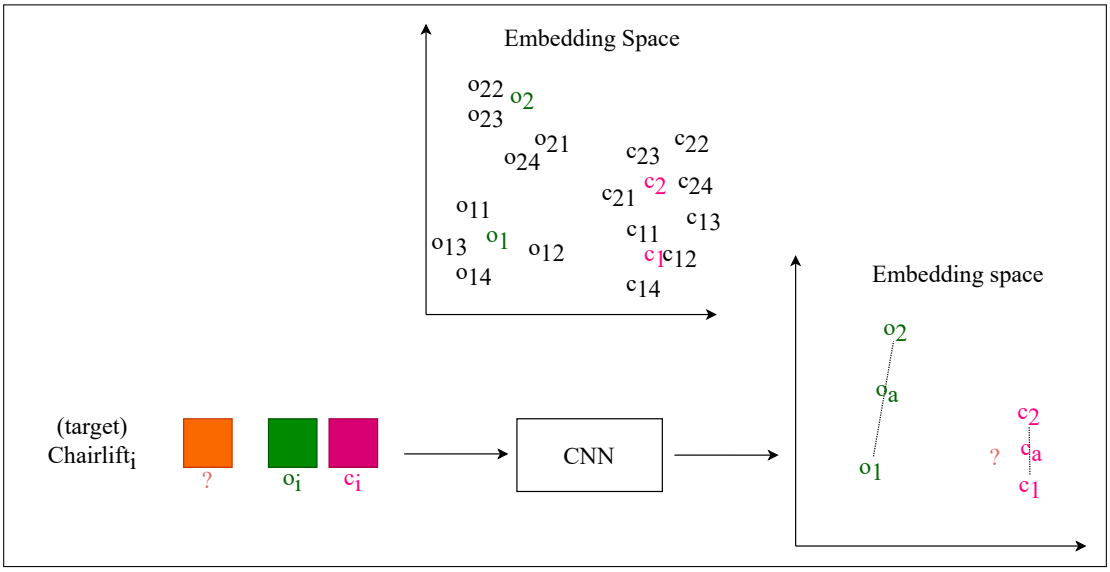


Figure 6.4 – Test on the target domain with average source masks as reference.

Testing on target data with target masks

In this section, we assume that the masks are available for each target chairlift. In this case, the solution is straightforward. We feed the network with the open and closed masks o_k, c_k corresponding to the k^{th} target domain chairlift as well as the test images I_k of this chairlift. Then, the Euclidean distances between the image feature vectors and the two mask feature vectors allow us to infer the class of the test image (see Figure 6.3):

$$\hat{Y} = \arg\min_{M_k \in \{o_k, c_k\}} \|y_{M_k} - y_{I_k}\|^2. \tag{6.4}$$

Where, $y_{I_k} = F(I_k)$ and $y_{M_k} = F(M_k)$ are the output vectors of the image and the mask, respectively.

This solution fully exploits the advantages of our learning process that allows to learn chairlift-dependent discriminators by comparing each image only with its corresponding masks while projecting all the feature vectors in the same embedding space, so that the features of different chairlifts can help to improve each other.

By looking at the feature vectors of the target open and closed images in the embedding space (see Figure 6.7), we notice that they are relatively far from the center of their corresponding cloud. This is expected because the model has never been trained on these images. Indeed, given the diversity of chairlift geometries, camera viewpoints, weather conditions and backgrounds, the target open (resp. closed) images can be far away from the average open (resp. closed) mask, even if they are near their real open (resp. closed) mask.

Consequently, after having the network trained on the source training images with the set of corresponding binary masks from N source chairlifts, as described in the previous section, we resort to a fine-tuning step by considering only the two target masks (see Figure 6.5).

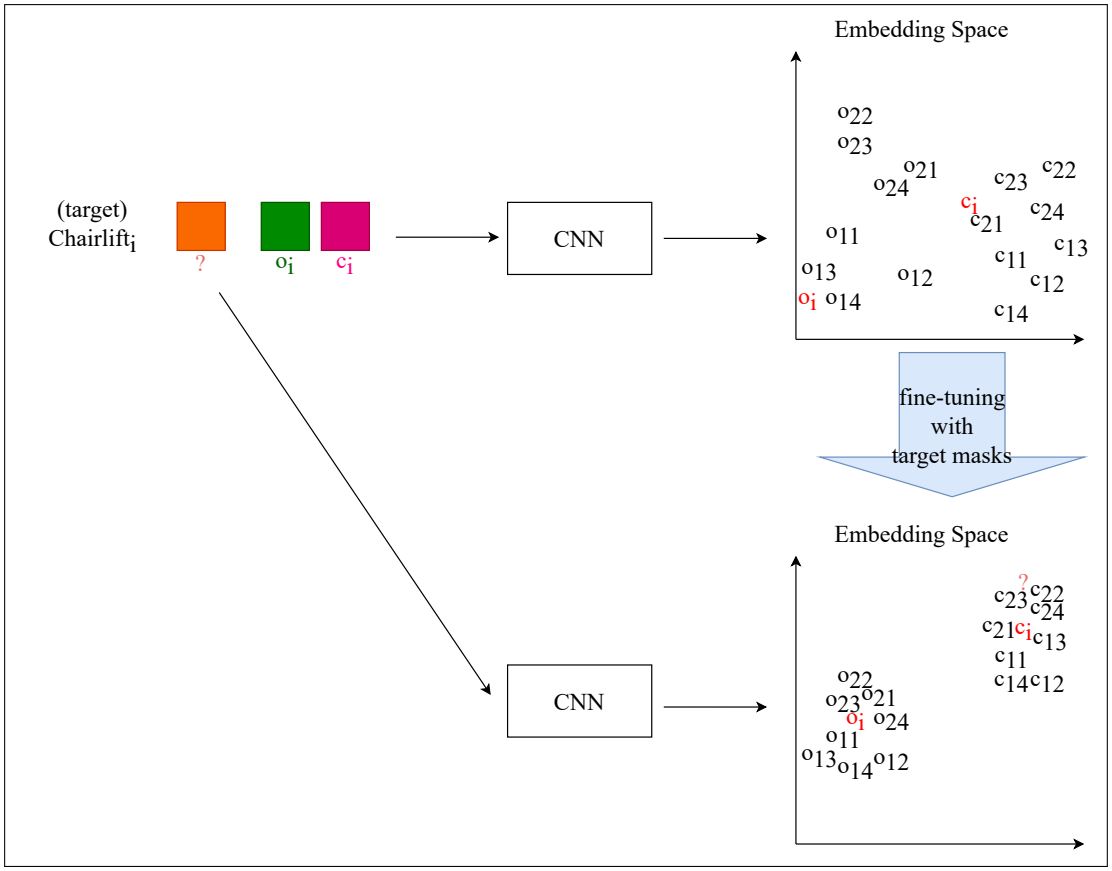


Figure 6.5 – Fine-tuning step with target masks.

More precisely, considering the training source images I_i and target masks M_k, and $y_{I_i} = F(I_i)$ and $y_{M_k} = F(M_k)$ their output vectors in the feature space, we train the Siamese model by minimizing the following contrastive loss:

$$\mathcal{L}(y_{I_i}, y_{M_k}) = \alpha \|y_{M_k} - y_{I_i}\|^2 + (1-\alpha)\max(1 - \|y_{M_k} - y_{I_i}\|, 0)^2 \tag{6.5}$$

where $M_k \in \{o_k, c_k\}$, $\|.\|$ denotes the L_2 norm, $\alpha = 1$ if the class of the image is the same as the class of the mask and $\alpha = 0$ otherwise.

After these two successive learning steps, the source open (resp. closed) images should concentrate around the target open (resp. closed) mask and far away the target closed (resp. open) mask. This is a property that improves the generalization power of the model. Indeed, the open (resp. closed) images from different chairlifts fall in the same area of the embedding space, even if they present noticeable differences in safety bar geometry, camera viewpoint, weather conditions and background. Thus, given an image I_k from the k^{th} target chairlift, its open mask o_k and closed mask c_k , we evaluate the distance between its feature vector and the two feature vectors of its masks. The nearest mask provides us the class of this test image using Equation 6.4.

A similar behavior (grouping all the open or closed images together) can be obtained by learning a simple binary classifier (open/closed) over the images of all the chairlifts, but we are showing in the experimental results that it does not provide as good results as our solution. Indeed, since a classical binary classifier is not designed to integrate geometry constraints such as the binary masks, it requires much more labeled training data to avoid over-fitting and provide accurate results on unseen domains. With our fine-tuning step, we take advantage of the strengths of both approaches: we are able to inform the classifier where it should focus by feeding the network with pairs of images and binary masks, making the learning step easier with few data; and we boost the generalization property of the model by introducing the target masks.

6.4 Experiments

To evaluate the efficiency of our approach, we conduct experiments in the context of MIVAO project, on the *tiny chairlifts dataset* explained in Section 5.4.1. This dataset has very few labeled data. Indeed, only 20 images per chairlift.

6.4.1 Experimental setup

Chairlifts datasets. We use the *tiny chairlifts dataset* which is composed of 21 different chairlifts. As we want to test our model ability to generalize, we consider the experimental settings *FS* ("Fifteen Six") explained in Section 3.3.2. In this setting, we randomly selected 15 chairlifts as source data and the 6 remaining chairlifts as target data (see Table 6.1). The training set is composed of images from the training sets of the 15 source chairlifts, and the test set is composed of the images of test sets of target chairlifts.

Table 6.1 – Distribution of the chairlifts between source and target domain in the tiny dataset.

Domain	data	chairlifts
Source	300 imgs + 30 masks	C_3, C_4,C_5,C_7,C_8,C_{10},C_{11},C_{13},C_{14},C_{15},C_{16},C_{17},C_{18},C_{19},C_{20}
Target	12 masks	C_0, C_1,C_2,C_6,C_9,C_{12}

Baseline network. The Siamese model has two identical networks that share weights. We propose to use the deep architecture composed of the convolutional part of VGG16 [153] pre-trained on IMAGENET [142] and two fully connected (FC) layers randomly initialized. The first FC layer has 4096 neurons and the second one has 1024 neurons. The baseline to this network is a classical binary classifier that uses the same architecture augmented with a two neurons classification layer. The number of parameters of this architecture is nearly the same as the one of the binary classifier, $94,411,584$ and $94,413,634$, respectively. All the networks are trained using back-propagation and the stochastic gradient descent algorithm. The maximum number of epochs is set to 1000. The learning rate is set to 10^{-5}, the learning rate decay to 10^{-8} and the momentum to 0.9. The inputs of our Siamese model are pairs of images constituted by one colored image of one chairlift and one of the two corresponding masks. During training, we make sure that the positive and negative pairs are well balanced so that we consider:

- 50% positive pairs: (open image - open mask) and (closed image - closed mask)
- 50% negative pairs: (open image - closed mask) and (closed image - open mask)

6.4.2 Results

Once the network has been trained on the 15 source chairlifts, we propose to test it on the 6 target chairlifts, which present very different geometries and acquisition conditions, as illustrated in Figure 6.1. Different settings are tested in this context and the results are shown in Table 6.2.

Table 6.2 – Accuracy of the the Siamese and the baseline classifier with deep architectures for all target chairlifts in the *tiny chairlifts dataset* using the configuration *FS*.

Target chairlift	Baseline classifier	Our Siamese model		
		Avg. source masks	Target masks	Target masks +fine-tuning
C_0	33.10	69.04	59.90	74.21
C_1	97.44	98.32	97.30	95.90
C_2	96.41	91.93	95.68	88.39
C_6	96.65	98.77	98.67	98.51
C_9	96.35	98.64	98.26	98.72
C_{12}	94.33	97.54	95.35	98.82
Avg.	85.71	92.37	90.86	92.43

First, we just evaluate the average masks of the source chairlifts and use them as references to test the target data. We note that this approach already outperforms the results from the baseline classifier (92.37% of accuracy on the target data, compared to 85.71%).

Second, we show the results provided when we use the binary target masks to classify the target images. This test confirms that inserting geometric constraints in a classification task is a good idea to provide the network a good generalization ability (90.86% of accuracy on the target data, compared to 85.71%).

Finally, we apply our fine-tuning step, which consists in bringing the target masks nearer to the corresponding source images cloud, as described above. We can see that this solution actually boosts the accuracy from 90.86% to 92.43% without using the target distribution. This shows that the proposed solution is accurate to remove the variations that occur between the different chairlifts.

It is worth mentioning here that C_0 is a special chairlift because it has a glass bubble for extra protection (see Figure 6.6), and it is the only chairlift that could be closed without being occupied, the data belongs to this chairlift is imbalanced, there is less than 3% of images in open class. And that explains the lower accuracy of the model archive on this particular chairlift.

Figure 6.6 – Examples of images of chairlift C_0. In the images of the first row, the chairlift carrier is empty, while in the second row, it is occupied. The safety bar is in different positions and so on the protection class (open or closed).

Comparing all methods

As this experiment has been performed previousely in the same setting to evaluate object detection in Chapter 3 and domain adaptation in Chapter 4 we can compare those methods to the Siamese model and the classical classifier. Table 6.3 sums up the results.

The baseline classifier (trained using image-level annotations) gave an accuracy of 85.71, while the object detector (trained and tested using instance-level annotations) achieved an accuracy of 87.12. Clearly, object location information improved image classification performance, but required bounding box annotations. Domain adaptation from the 15 source chairlifts to the 6 target chairlifts, enhanced the accuracy from 87.12 to 90.20. In this case, in addition to the 300 labeled training images from the source domain we needed 120 unlabeled images from the target domain. Our Siamese model achieved an outstanding accuracy of 92.34 without using the any information from the target domain in training, nor bounding box annotation. It only required 300 training images and 30 masks from the source domain. These results clearly show that the proposed Siamese model is a very good solution to insert shape prior in the classification model

without increasing its complexity. This allows the network to be trained with few labelled data and to well generalize on a new unseen data

Table 6.3 – Accuracy of different approaches with VGG16 backbone for all target chairlifts in the *tiny chairlifts dataset* using the configuration *FS*.

Approach	Accuracy
Image classification	85.71
Faster R-CNN	87.12
Faster R-CNN with domain adaptation	90.20
Siamese model	92.43

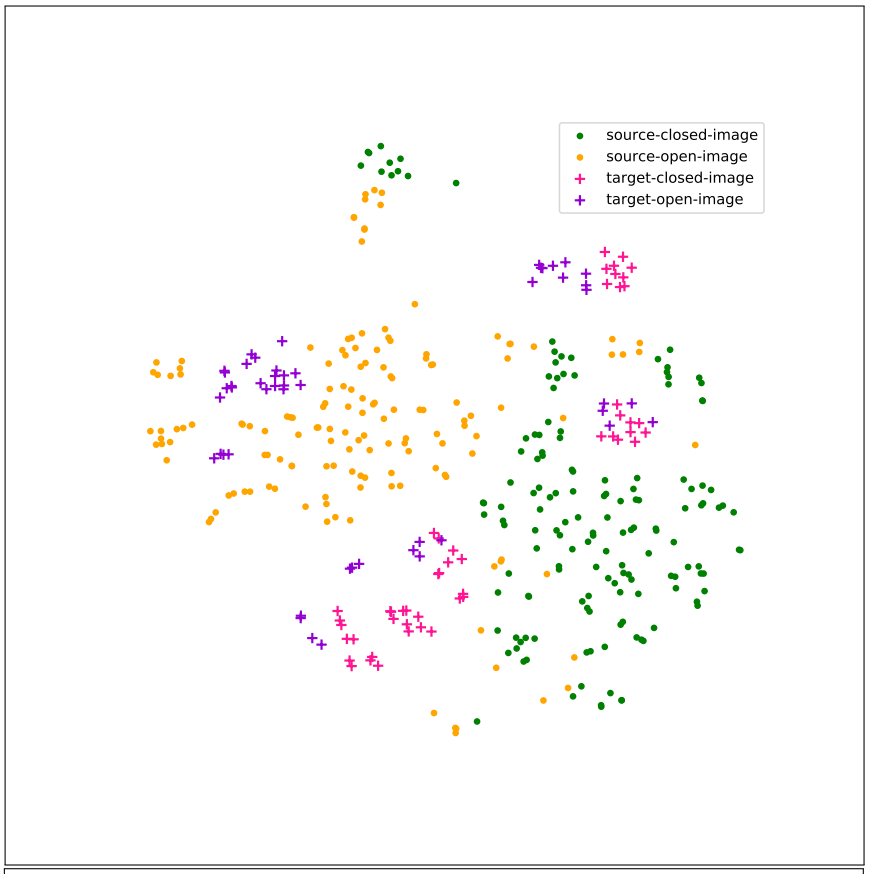
Observing the embedding space

To observe the distribution of the data in the embedding space, we propose to apply the tSNE tool (t-distributed Stochastic Neighbor Embedding) [157] on our data. The interesting part is that we can also project the feature vectors of the last layer of the baseline classifier to compare it with the Siamese distribution.

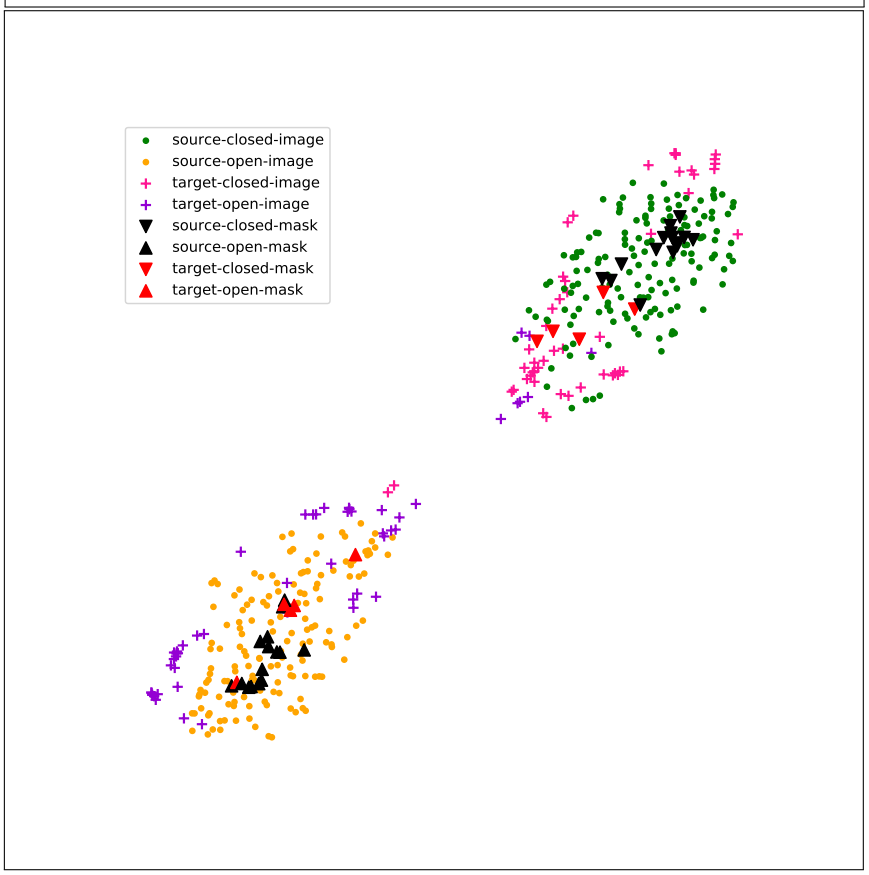
Figure 6.7 shows the embedding space of both the baseline classifier and the Siamese model. We can see on the two plots both samples from the source data that have been used to learn the network depicted with dot markers and samples from the target data depicted with crosses. The colors indicate the class of each point. In these plots, we can clearly see that the baseline classifier is well separating the two classes when considering the source data, but we can also see that the target data is sometimes not on the correct side of the classifier. On the other side, we can see that the Siamese architecture moves the points of different classes away from each other thanks to the contrastive loss and that the target points are also mainly following the source distribution. We also notice that the open (resp. closed) masks are near each other and far from the closed (resp. open) masks. We do not use any constraint to enforce this behavior, but since we are learning on a set of different chairlifts and project them in the same embedding space, the model is automatically learning the features that help to discriminate the open masks from the closed masks.

6.5 Conclusion

This chapter has addressed a challenging case of image classification, where few labeled images are available, and generalization over unseen target domains is needed. Our proposition relies on very intuitive ideas implemented in the framework of a specific embedding space with Euclidean properties. Supposing each class comes with a binary mask focusing on relevant elements to detect, the embedding space is first learned over a set of known source domains with a Siamese model with image-mask pairs as input. A contrastive loss is used to provide Euclidean properties to the space. To provide good generalization properties to the model to classify images of an unseen domain, we use two virtual masks computed in the embedding space as the centroid of the source masks. In the context of the chairlift safety problem, we have shown that this approach performs much better than a generic image classifier in the target domain. Finally, if target masks are available, we have proposed a simple fine-tuning step that helps to improve the generalization property of the model, by concentrating images from different chairlifts around the same binary masks. At test time, the classification can be achieved using a simple Euclidean distance in the original embedding space.



(a)



(b)

Figure 6.7 – Source and target images and masks visualised using TSNE. (a) Embedding space of the baseline classifier, (b) Embedding space of the Siamese model with the corresponding open and closed masks. See text for explanation.

Conclusion

This thesis presents our work on the chairlift safety problem using deep learning techniques. As part of Mivao project, we aim to ensure safety in chairlifts by monitoring the boarding station and detecting risky situations to take corrective actions before an accident occurs. Given an image showing people boarding a chairlift, we have proposed a set of solutions to categorize the situation as safe or unsafe. Since there is no public dataset containing such specific images, we had to propose models that can be trained with few annotated data. Furthermore, two main issues had to be addressed for this project. First, since the safety property is highly related to the position of the safety bar, the proposed deep models should concentrate on small but crucial areas in the images. Second, due to the difficulty of labeling new images when a system is installed on a new unseen chairlift, the models trained on a set of chairlifts should provide good results on a new chairlift without labeled images.

The latest advances in the academic community have the potential to be applied to our problem despite their limitations. In Chapter 2, we described some related state of the art methods that could be applied in our context, mainly in the fields of image classification, object detection, domain adaptation, and guided problems under constraints. Research developments in these research areas have raised expectations, especially in the age of deep learning, which has improved significantly in the last few years. However, very few works have attempted to use a deep learning-based system in chairlift security by computer vision. Moreover, the main limitations of standard deep learning methods in our context are (i) they require massive sets of annotated images for training, which is not always available in our case, (ii) it is not easy to determine which elements of the scene are used for the prediction, and (iii) they can have low generalization properties if there is a shift between the distributions of source and target domains. Object detection is a promising approach to chairlift safety, which allows to focus on specific local features in the scene. Furthermore, there are some good solutions for unsupervised domain adaptation, to enhance the ability of the object detector trained on a source domain to perform well on a new unseen domain. On the other hand, all the current adaptive object detectors are not adapted at the proposals-level. Finally, the Siamese network is a convenient and efficient method of exploiting prior knowledge in the training phase.

Our first contribution is to show that an object detection approach can outperform a classification one. Chapter 3 was dedicated to study the performance and limits of an object detector to solve the chairlift safety problem. We found that object detection helps the network to concentrate on crucial details in the images. Object location information can be beneficial for further pushing image classification, particularly in generalization. Furthermore, when a sufficient amount of data is available, Faster R-CNN provides high performance. However, the performance is affected when the test set is a new domain with a divergence from the training domain, indicating the need for a domain adaptation step. Another strength of object detection is its ability to detect some essential elements in the scene, such as adults and children. This additional information can help to develop a better understanding of the scene to ultimately categorize it into safe or unsafe.

Our second contribution is to propose a domain adaptation component at the region proposal level of a two-stage detector. Chapter 4 focused on improving the generalization ability of an object detector on a new target domain. State of the art domain adaptation solutions for two-stage object detection mainly concentrate on the global and local levels, neglecting the region proposal branch. We have shown that adaptation is required at the region proposal level to provide better detection results in the target domain where no labels are available, and this original solution outperformed the classical solutions.

Our third contribution is to propose a Siamese model to insert some geometric priors in an image classification network. Chapter 5 was oriented towards a new approach, forcing an image classification network to look at regions likely to contain the target object using binary masks without the need for instance-level bounding box annotations. We have shown that by using a Siamese architecture, the network could attend to the most crucial elements of the scene. Our Siamese network took as input a pair of images: one colored image of the scene, and one binary image representing the mask of the safety bar. Only two masks were required for each chairlift. This approach proved its efficiency even with few labeled data and outperformed a classical image classifier.

Our final contribution is a set of methods to asses and boost the generalization property of the Siamese model. Chapter 6 continued on the same thread of Chapter 5 and aimed to evaluate the generalizability of our mask-guided approach by testing our Siamese architecture on new unseen chairlifts with different 3D geometries. This chapter showed that our Siamese network had high generalization properties, indicating that as the domains of interest expand, the focus on the most crucial visual regions in the scene became more important. A single model trained on a set of chairlifts provided outstanding results on a new unseen chairlift, without requiring any adaptation step. It is worth noting that this last model outperformed the adaptive object detector.

The chairlift safety problem and some of its possible solutions from a computer vision perspective have been presented and discussed in this thesis. However, we believe that this work deserves to be further developed in the future in order to overcome its limitations or use it in new applications. There are many potential directions to improve our various contributions that have yet to be explored. Following, some possible future works are discussed. First of all, Faster R-CNN results could be enhanced using advanced deep architectures to extract features like ResNet. There are many robust architectures available today, and they are potentially much better or faster than the ones we have tested, but we could not test them all because of time constraints, and also because new deep learning models are continually being developed, which makes it challenging to keep track of the state of the art.

Faster R-CNN is proven to be weak in classifying the detected objects of certain types, as shown in Chapter 3: 'child' objects tend to be particularly challenging. We need to continue exploring effective object detection algorithms, analyzing the latest frameworks (especially one-stage detectors), and studying the possibility to use domain adaptation in these frameworks. One perspective is to evaluate the detection of people and safety bars in a unified framework, which was not possible because we did not have annotated version of the dataset with all classes. Then, we can compare these results with those obtained by our partners in the project, since we will be able to classify safe and unsafe situations instead of only detecting the essential elements of the scene.

Chapter 4 showed that using domain adaptation for object detection improved the results, but adapting the models of a source domain to a target domain is not the ideal solution because the success of this adaptation strongly depends on the similarity between these two particular domains. It is therefore more efficient to rather focus on domain generalization from a set of source domains to a target domain. Also, more accurate adaptation procedures could also improve the results. These methods could help in the learning step to reach stable solutions which is a substantial weakness of the current methods.

In attempting to use masks with few annotated data in Chapter 5, despite its efficiency, Siamese architecture is slower than its corresponding baseline because the data is double, so a strategy for selecting a few pairs to speed up the training while maintaining the same accuracy would be necessary. Moreover, the same point about comparing different backbones to improve results could be applied here as well. The shallow architecture was promising, so it is worth optimizing it by changing number of layers, neurons in each layer and filters sizes.

Finally, in the Chapter 6 Siamese model provided an exceptional generalization power without using any information from the target domain. We think that a new boosting process using only the virtual masks would further improve generalization. Another perspective to validate the performance of this approach would be to test it in another application different from chairlift safety.

We tackled the problem of chairlift safety in both object detection and image classification. One natural next question is: would it be worth to analyze video rather than image data? Would it be advantageous to employ temporal information and tackle the problem as object tracking?

Bibliography

[1] Bryan Scott Queen. Systems and methods for improved operations of ski lifts, April 21 2020. US Patent 10,628,679.

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

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[4] STRMTG. `http://www.strmtg.developpement-durable.gouv.fr/accidents-a249.html`. Accessed: 2021-05-16.

[5] G Ruedl, A Schranz, C Fink, E Pocecco, W Nachbauer, and M Burtscher. Are acl injuries related to perceived fatigue in female skiers? In *Skiing Trauma and Safety, 18th Volume*. ASTM International, 2011.

[6] Michael Cusimano, Wilson P Luong, Ahmed Faress, Timothy Leroux, and Kelly Russell. Evaluation of a ski and snowboard injury prevention program. *International journal of injury control and safety promotion*, 20(1):13–18, 2013.

[7] Matthias Gilgien, Jörg Spörri, Josef Kröll, and Erich Müller. Effect of ski geometry and standing height on kinetic energy: equipment designed to reduce risk of severe traumatic injuries in alpine downhill ski racing. *British journal of sports medicine*, 50(1):8–13, 2016.

[8] Annabelle Davey, Nathan K Endres, Robert J Johnson, and Jasper E Shealy. Alpine skiing injuries. *Sports health*, 11(1):18–26, 2019.

[9] Lauren A Pierpoint, Zachary Y Kerr, Gary Grunwald, Morteza Khodaee, Tessa Crume, and R Dawn Comstock. Effect of environmental conditions on injury rates at a colorado ski resort. *Injury prevention*, 2019.

[10] Boris Delibašić, Sandro Radovanović, Miloš Z Jovanović, and Milija Suknović. Improving decision-making in ski resorts by analysing ski lift transportation—a review. In *Advances in operational research in the balkans*, pages 265–273. Springer, 2020.

[11] Sandro Radovanovic, Boris Delibasic, Milija Suknovic, and Dajana Matovic. Where will the next ski injury occur? a system for visual and predictive analytics of ski injuries. *Operational Research*, 19(4):973–992, 2019.

[12] Vassilios Tsakanikas and Tasos Dagiuklas. Video surveillance systems-current status and future trends. *Computers & Electrical Engineering*, 70:736–753, 2018.

[13] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48, 2012.

[14] In Su Kim, Hong Seok Choi, Kwang Moo Yi, Jin Young Choi, and Seong G Kong. Intelligent visual surveillance—a survey. *International Journal of Control, Automation and Systems*, 8(5):926–939, 2010.

[15] Maria Valera and Sergio A Velastin. Intelligent distributed surveillance systems: a review. *IEE Proceedings-Vision, Image and Signal Processing*, 152(2):192–204, 2005.

[16] Fereshteh Falah Chamasemani, Lilly Suriani Affendey, et al. Systematic review and classification on video surveillance systems. *International Journal of Information Technology and Computer Science (IJITCS)*, 5(7):87, 2013.

[17] Tanin Sultana and Khan A Wahid. Iot-guard: Event-driven fog-based video surveillance system for real-time security management. *IEEE Access*, 7:134881–134894, 2019.

[18] Arie Hans Nasution and Sabu Emmanuel. Intelligent video surveillance for monitoring elderly in home environments. In *2007 IEEE 9th Workshop on Multimedia Signal Processing*, pages 203–206. IEEE, 2007.

[19] VC Maha Vishnu, M Rajalakshmi, and R Nedunchezhian. Intelligent traffic video surveillance and accident detection system with dynamic traffic signal control. *Cluster Computing*, 21(1):135–147, 2018.

[20] Sohail Salim, Othman O Khalifa, Farah Abdul Rahman, and Adidah Lajis. Crowd detection and tracking in surveillance video sequences. In *2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, pages 1–6. IEEE, 2019.

[21] Dae Hyun Ryu, HyungJun Kim, and Keehong Um. Reducing security vulnerabilities for critical infrastructure. *Journal of Loss Prevention in the Process Industries*, 22(6):1020–1024, 2009.

[22] Ying Meng and Hongtao Wu. Highway visibility detection method based on surveillance video. In *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, pages 197–202. IEEE, 2019.

[23] Francisco Manuel Castro, Rubén Delgado-Escaño, Nicolás Guil, and Manuel Jesús Marín-Jiménez. A weakly-supervised approach for discovering common objects in airport video surveillance footage. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 296–308. Springer, 2019.

[24] H Wei, M Laszewski, and N Kehtarnavaz. Deep learning-based person detection and classification for far field video surveillance. In *2018 IEEE 13th Dallas Circuits and Systems Conference (DCAS)*, pages 1–4. IEEE, 2018.

[25] Mohamed Elhoseny. Multi-object detection and tracking (modt) machine learning model for real-time video surveillance systems. *Circuits, Systems, and Signal Processing*, 39(2):611–630, 2020.

[26] Apoorva Raghunandan, Pakala Raghav, HV Ravish Aradhya, et al. Object detection algorithms for video surveillance applications. In *2018 International Conference on Communication and Signal Processing (ICCSP)*, pages 0563–0568. IEEE, 2018.

[27] Wei Lei, Dongjun Huang, and Xiwen Cui. Moving object tracking in video surveillance using yolov3 and meanshift. In *Tenth International Conference on Graphics and Image Processing (ICGIP 2018)*, volume 11069, page 1106940. International Society for Optics and Photonics, 2019.

[28] Issam Elafi, Mohamed Jedra, and Noureddine Zahid. Unsupervised detection and tracking of moving objects for video surveillance applications. *Pattern Recognition Letters*, 84:70–77, 2016.

[29] Muhammad Attique Khan, Kashif Javed, Sajid Ali Khan, Tanzila Saba, Usman Habib, Junaid Ali Khan, and Aaqif Afzaal Abbasi. Human action recognition using fusion of multiview and deep features: an application to video surveillance. *Multimedia Tools and Applications*, pages 1–27, 2020.

[30] Haoran Wei and Nasser Kehtarnavaz. Semi-supervised faster rcnn-based person detection and load classification for far field video surveillance. *Machine Learning and Knowledge Extraction*, 1(3):756–767, 2019.

[31] Joey Tianyi Zhou, Jiawei Du, Hongyuan Zhu, Xi Peng, Yong Liu, and Rick Siow Mong Goh. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, 14(10):2537–2550, 2019.

[32] Serhan Coşar, Giuseppe Donatiello, Vania Bogorny, Carolina Garate, Luis Otavio Alvares, and François Brémond. Toward abnormal trajectory and event detection in video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):683–695, 2016.

[33] Rémi Dufour, Cyril Meurie, and Amaury Flancquart. Annotation tool designed for hazardous user behavior in guided mountain transport. In *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, pages 162–168. IEEE, 2018.

[34] Kevin Bascol, Rémi Emonet, Elisa Fromont, and Raluca Debusschere. Improving chairlift security with deep learning. In *International Symposium on Intelligent Data Analysis*, pages 1–13. Springer, 2017.

[35] Kevin Bascol, Rémi Emonet, and Elisa Fromont. Improving domain adaptation by source selection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3043–3047. IEEE, 2019.

[36] Julien Muzeau, Patricia Ladret, and Pascal Bertolino. Linear classification of chairlift images for presence analysis. In *Fourteenth International Conference on Quality Control by Artificial Vision*, volume 11172, page 1117205. International Society for Optics and Photonics, 2019.

[37] Hiba Alqasir, Damien Muselet, and Christophe Ducottet. Region proposal oriented approach for domain adaptive object detection. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 38–50. Springer, 2020.

[38] Hiba Alqasir, Damien Muselet, and Christophe Ducottet. Mask-guided image classification with siamese networks. In *International Conference on Computer Vision Theory and Applications*, 2020.

[39] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[40] Bahram Javidi. *Image recognition and classification: algorithms, systems, and applications*. CRC press, 2002.

[41] Rajkumar Buyya, Rodrigo N Calheiros, and Amir Vahid Dastjerdi. *Big data: principles and paradigms*. Morgan Kaufmann, 2016.

[42] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[43] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[44] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[45] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989.

[46] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[47] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[48] Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4749–4757, 2015.

[49] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[50] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[51] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[52] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

[53] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[54] Chen-Lin Zhang, Jian-Hao Luo, Xiu-Shen Wei, and Jianxin Wu. In defense of fully connected layers in visual representation transfer. In *Pacific Rim Conference on Multimedia*, pages 807–817. Springer, 2017.

[55] Jun-e Liu and Feng-Ping An. Image classification algorithm based on deep learning-kernel function. *Scientific Programming*, 2020, 2020.

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

[57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[58] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a backpropagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[59] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[61] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014.

[63] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[64] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[65] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, page 4, 2017.

[66] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[67] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[68] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[69] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[70] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[71] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. Dsod: Learning deeply supervised object detectors from scratch. In *The IEEE International Conference on Computer Vision (ICCV)*, page 7, 2017.

[72] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 821–830, 2019.

[73] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

[74] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–2202, 2012.

[75] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[76] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2154, 2014.

[77] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[78] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2974, 2019.

[79] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018.

[80] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9657–9666, 2019.

[81] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 6054–6063, 2019.

[82] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[83] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[84] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8577–8584, 2019.

[85] Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Hierarchical shot detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9705–9714, 2019.

[86] Jing Nie, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Enriched feature guided refinement network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9537–9546, 2019.

[87] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Consistent optimization for single-shot object detection. *arXiv preprint arXiv:1901.06563*, 2019.

[88] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018.

[89] Hongkai Zhang, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cascade retinanet: Maintaining consistency for single-stage object detection. *arXiv preprint arXiv:1907.06881*, 2019.

[90] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019.

[91] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019.

[92] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[93] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020.

[94] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[95] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[96] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[97] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[98] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[99] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[100] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[101] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

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

[103] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pages 3536–3544, 2014.

[104] Anant Raj, Vinay P Namboodiri, and Tinne Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. *arXiv preprint arXiv:1507.05578*, 2015.

[105] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5001–5009, 2018.

[106] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3339–3348, 2018.

[107] Yushan Yu, Xuemiao Xu, Xiaowei Hu, and Pheng-Ann Heng. Dalocnet: Improving localization accuracy for domain adaptive object detection. *IEEE Access*, 7:63155–63163, 2019.

[108] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.

[109] Yuhu Shan, Wen Feng Lu, and Chee Meng Chew. Pixel and feature level based domain adaptation for object detection in autonomous driving. *Neurocomputing*, 367:31–38, 2019.

[110] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019.

[111] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 749–757, 2020.

[112] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013.

[113] Kaiyang Zhou, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pages 13025–13032, 2020.

[114] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018.

[115] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008.

[116] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[117] Yongxin Yang and Timothy Hospedales. A unified perspective on multi-domain and multi-task learning. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

[118] Atsutoshi Kumagai and Tomoharu Iwata. Zero-shot domain adaptation without domain semantic descriptors. *ArXiv*, abs/1807.02927, 2018.

[119] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. Link the head to the" beak": Zero shot learning from noisy text description at part precision. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6288–6297. IEEE, 2017.

[120] Richard A Caruana. Multitask connectionist learning. In *In Proceedings of the 1993 Connectionist Models Summer School*. Citeseer, 1993.

[121] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[122] Wonhee Lee, Joonil Na, and Gunhee Kim. Multi-task self-supervised object detection via recycling of bounding box annotations. In *2019 IEEE conference on computer vision and pattern recognition (CVPR)*. Ieee, 2019.

[123] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir Rawashdeh. Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2019.

[124] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[125] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[126] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

[127] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[128] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*, 2015.

[129] Xiangyun Zhao, Shuang Liang, and Yichen Wei. Pseudo mask augmented object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4061–4070, 2018.

[130] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3992–4000, 2015.

[131] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *2018 IEEE conference on computer vision and pattern recognition (CVPR)*. Ieee, 2018.

[132] Yanwei Pang, Jin Xie, Muhammad Haris Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Mask-guided attention network for occluded pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4967–4975, 2019.

[133] Dingwen Zhang, Huazhu Fu, Junwei Han, Ali Borji, and Xuelong Li. A review of co-saliency detection technique: Fundamentals, applications, and challenges. *arXiv preprint arXiv:1604.07090*, 2016.

[134] Kaihua Zhang, Tengpeng Li, Bo Liu, and Qingshan Liu. Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3095–3104, 2019.

[135] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 398–407, 10 2017.

[136] J. Zhao and C. G. M. Snoek. Dance with flow: Two-in-one stream action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[137] 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*, pages 850–865. Springer, 2016.

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

[139] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4010–4019, 2019.

[140] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020.

[141] Sovan En, Alexis Lechervy, and Frédéric Jurie. Ts-net: combining modality specific and common features for multimodal patch matching. In *2018 IEEE International Conference on Image Processing (ICIP)*. Ieee, 2018.

[142] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[143] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[144] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[145] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[146] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[147] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

[148] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[149] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[150] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, pages 1–20, 2018.

[151] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A. Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *International Conference on Learning Representations*, 2018.

[152] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.

[153] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[154] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[155] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[156] Kuan-Chuan Peng, Ziyan Wu, and Jan Ernst. Zero-shot deep domain adaptation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[157] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Appendix A

Additional results

The first experiments carried out in this thesis, consist in applying Faster R-CNN on our chairlifts dataset to evaluate object detection in the chairlift safety problem. We formulated the problem as object detection of the important elements of the scene *i.e.* the safety bar and the people. In Chapter 3 we demonstrated general results, and in this appendix we show more detailed ones.

A.1 Safety bar detection

In section 3.3 we provided the implementation details of this experiment and average results over all the chairlifts. In this section, we show detailed results for each chairlift independently in the four settings *i.e.* *OOC*, *LOCO*, *All* and *FS*.

A.1.1 OOC

ZF Backbone

Faster R-CNN produces 0.95 mAP in *OOC* setting using ZF feature extractor, which implies that when trained on images belongs to a chairlift the detector is able to detect with height precision and recall most of the objects in images from the same chairlift. Note that in this setting the dataset is small and with too few variations. Table A.1 shows the results for each chairlift independently, the detector produces relatively bad results only on one chairlift C_7 with mAP equals to 0.62 that is because of the great imbalance in the data between the two classes in this chairlift see Table 1.2, C_7 has less than 10% of images in closed class.

The detector produces fairly good results on two chairlifts with mAP equals to 0.85 or 0.86 (C_0, C_2 respectively), it is worth mentioning here that C_0 is a special chairlift because it has a glass bubble for extra protection and it is the only chairlift that could be closed without being occupied, the data belongs to this chairlift is imbalanced, there is less than 3% of images in open class.

Perfect results were achieved on eight chairlifts with mAP equals to 0.99 or 1 (C_3, C_5, C_6, C_9, C_{13}, C_{14}, C_{15}, C_{16}) most of these chairlifts images are well balanced between the two classes.

While the detector achieved very good results with mAP between 0.95 and 0.98 on the rest of the chairlifts (C_1, C_4, C_8, C_{10}, C_{11}, C_{12}, C_{17}, C_{18}, C_{19}, C_{20}) most of these chairlifts images are not very good balanced between the two classes. The unbalanced distribution of classes is detrimental, when only a few examples of a certain class used during the training then the results of this class will be affected.

VGG16 Backbone

The mAP achieved by Faster R-CNN when VGG16 used as feature extractor increased to 0.99 (0.04 better than ZF feature extractor). Table A.2 shows the results for each chairlift independently, the mAP is higher than 0.97 for all chairlifts. The class imbalance is not a problem using this backbone. Even the special chairlift C_0 has better results.

A.1.2 All

ZF Backbone

As expected the best results has been achieved in this setting, because of the huge size of the dataset in one hand, and all the domain are well-represented with considerable variations in the other hand. Table A.3 shows Faster R-CNN with ZF backbone results for each chairlift independently, note that the results of C_7, C_2 are significantly improved than they were in OOC setting using the same backbone. While the special case C_0 has reduced mAP and that is because it is not well represented.

VGG16 Backbone

The mAP achieved by Faster R-CNN when VGG16 used as feature extractor slightly increased to 0.99 (0.01 better than ZF feature extractor). Table A.4 shows the results for each chairlift independently. The mAP of the special case C_0 has considerably improved, but is still not better than the mAP obtained using the same backbone in the OOC setting.

A.1.3 LOCO

ZF Backbone

In $LOCO$ experiment with ZF backbone, the performance of both classes decreased for all the measures, mAP for the different chairlifts has a wide range between 0.15 and 0.9 (see Table A.5). Which is expected because in this setting, the images used in test belongs to a chairlift that is completely unseen before. If the safety bar of this chairlift is similar to other safety bars well represented in the dataset there is a good chance to have good results. Where if it is completely different, or similar to a safety bar which is not well represented in the dataset, that well harm the performance.

VGG16 Backbone

Using VGG16 backbone in $LOCO$ experiments reduce the harm in performance, with mAP equal to 0.84 (0.25 better than ZF backbone). The detector easily profited from the deeper and more expressive features of the VGG16.

Table A.1 – Safety bar detection (by chairlift) results in *OOC* setting, using Faster R-CNN model with ZF backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	closed	0.95	0.95	0.95	0.93	0.85
	open	0.94	0.80	0.86	0.77	
C_1	closed	0.96	0.98	0.97	0.97	0.97
	open	1.00	0.97	0.98	0.97	
C_2	closed	0.82	0.82	0.82	0.73	0.86
	open	0.99	0.99	0.99	0.98	
C_3	closed	0.97	1.00	0.99	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_4	closed	0.95	0.95	0.95	0.93	0.97
	open	1.00	1.00	1.00	1.00	
C_5	closed	0.98	0.99	0.99	0.99	0.99
	open	0.99	0.99	0.99	0.98	
C_6	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_7	closed	0.88	0.91	0.90	0.88	0.62
	open	0.57	0.57	0.57	0.35	
C_8	closed	0.95	0.95	0.95	0.92	0.95
	open	0.98	0.99	0.99	0.97	
C_9	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{10}	closed	0.97	0.98	0.97	0.97	0.98
	open	1.00	0.99	0.99	0.99	
C_{11}	closed	0.98	0.98	0.98	0.98	0.97
	open	0.98	0.97	0.97	0.95	
C_{12}	closed	1.00	1.00	1.00	1.00	0.97
	open	0.96	0.96	0.96	0.94	
C_{13}	closed	0.99	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{14}	closed	0.99	0.99	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_{15}	closed	0.99	0.99	0.99	0.99	0.99
	open	0.99	0.99	0.99	0.98	
C_{16}	closed	0.99	1.00	0.99	0.99	0.99
	open	1.00	0.99	0.99	0.99	
C_{17}	closed	0.94	0.94	0.94	0.92	0.95
	open	0.99	0.99	0.99	0.98	
C_{18}	closed	0.98	0.98	0.98	0.98	0.98
	open	0.99	0.99	0.99	0.98	
C_{19}	closed	0.98	0.99	0.98	0.98	0.97
	open	0.99	0.97	0.98	0.96	
C_{20}	closed	0.99	0.99	0.99	0.98	0.98
	open	0.99	0.99	0.99	0.98	
Avg..	closed	0.96	0.97	0.97	0.96	0.95
	open	0.97	0.96	0.96	0.94	

Table A.2 – Safety bar detection (by chairlift) results in *OOC* setting, using Faster R-CNN model with VGG16 backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	closed	1.00	1.00	1.00	0.99	0.98
	open	1.00	0.97	0.98	0.97	
C_1	closed	0.98	0.98	0.98	0.97	0.98
	open	1.00	1.00	1.00	1.00	
C_2	closed	0.99	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_3	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_4	closed	0.96	0.96	0.96	0.94	0.97
	open	1.00	1.00	1.00	1.00	
C_5	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_6	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_7	closed	0.99	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_8	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_9	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{10}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{11}	closed	0.99	0.99	0.99	0.98	0.98
	open	1.00	0.99	0.99	0.99	
C_{12}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{13}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{14}	closed	0.99	0.99	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_{15}	closed	1.00	1.00	1.00	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{16}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{17}	closed	0.98	0.98	0.98	0.97	0.98
	open	1.00	1.00	1.00	1.00	
C_{18}	closed	0.99	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{19}	closed	0.99	0.99	0.99	0.98	0.97
	open	1.00	0.97	0.99	0.97	
C_{20}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
Avg.	closed	0.99	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	

Table A.3 – Safety bar detection (by chairlift) results in the setting *All*, using Faster R-CNN model with ZF backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	closed	1.00	1.00	1.00	1.00	0.67
	open	0.57	0.5	0.53	0.33	
C_1	closed	0.99	0.99	0.99	0.98	0.98
	open	1.00	0.98	0.99	0.98	
C_2	closed	0.91	0.91	0.91	0.89	0.95
	open	1.00	1.00	1.00	1.00	
C_3	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_4	closed	0.96	0.96	0.96	0.93	0.97
	open	1.00	1.00	1.00	1.00	
C_5	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_6	closed	1.00	1.00	1.00	1.00	0.99
	open	1.00	0.98	0.99	0.98	
C_7	closed	0.88	0.88	0.88	0.81	0.90
	open	1.00	0.99	1.00	0.99	
C_8	closed	1.00	0.95	0.97	0.95	0.97
	open	1.00	1.00	1.00	1.00	
C_9	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{10}	closed	1.00	0.97	0.99	0.97	0.98
	open	1.00	1.00	1.00	1.00	
C_{11}	closed	0.99	0.99	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_{12}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{13}	closed	1.00	0.99	1.00	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{14}	closed	0.98	0.98	0.98	0.96	0.97
	open	1.00	0.99	0.99	0.99	
C_{15}	closed	1.00	0.98	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_{16}	closed	1.00	1.00	1.00	1.00	0.99
	open	1.00	0.99	1.00	0.99	
C_{17}	closed	0.98	0.95	0.96	0.95	0.97
	open	1.00	0.99	1.00	0.99	
C_{18}	closed	1.00	0.97	0.99	0.97	0.98
	open	1.00	0.99	0.99	0.99	
C_{19}	closed	0.98	0.98	0.98	0.97	0.96
	open	1.00	0.96	0.98	0.96	
C_{20}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
Avg.	closed	0.99	0.98	0.99	0.98	0.98
	open	1.00	0.99	1.00	0.99	

Table A.4 – Safety bar detection (by chairlift) results in the setting *All*, using Faster R-CNN model with VGG16 backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	closed	0.99	0.99	0.99	0.99	0.93
	open	1.00	0.87	0.93	0.87	
C_1	closed	0.99	0.99	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_2	closed	0.99	0.99	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_3	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_4	closed	0.96	0.96	0.96	0.94	0.97
	open	1.00	1.00	1.00	1.00	
C_5	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_6	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_7	closed	0.96	0.96	0.96	0.95	0.97
	open	1.00	1.00	1.00	1.00	
C_8	closed	1.00	0.99	1.00	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_9	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{10}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{11}	closed	0.99	0.99	0.99	0.98	0.98
	open	1.00	1.00	1.00	0.99	
C_{12}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_{13}	closed	1.00	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{14}	closed	0.99	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{15}	closed	1.00	0.99	1.00	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{16}	closed	1.00	1.00	1.00	1.00	0.99
	open	1.00	0.99	1.00	0.99	
C_{17}	closed	0.98	0.98	0.98	0.97	0.98
	open	1.00	1.00	1.00	1.00	
C_{18}	closed	1.00	0.98	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_{19}	closed	0.99	0.99	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_{20}	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
Avg.	closed	0.99	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	

Table A.5 – Safety bar detection (by chairlift) results in *LOCO* setting, using Faster R-CNN model with ZF backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	closed	0.41	0.21	0.28	0.15	0.08
	open	0.01	0.22	0.02	0.01	
C_1	closed	0.79	0.84	0.82	0.76	0.76
	open	0.89	0.83	0.86	0.76	
C_2	closed	0.01	0.01	0.01	0.00	0.29
	open	0.76	0.75	0.76	0.58	
C_3	closed	0.53	0.80	0.63	0.63	0.40
	open	0.41	0.38	0.40	0.16	
C_4	closed	0.99	1.00	0.99	0.99	0.90
	open	0.90	0.90	0.90	0.81	
C_5	closed	0.99	1.00	0.99	0.99	0.90
	open	0.90	0.90	0.90	0.81	
C_6	closed	0.90	0.92	0.91	0.88	0.90
	open	0.97	0.95	0.96	0.92	
C_7	closed	0.12	0.62	0.20	0.29	0.15
	open	0.00	0.00	0.00	0.00	
C_8	closed	0.37	0.93	0.53	0.84	0.44
	open	0.30	0.09	0.13	0.03	
C_9	closed	0.82	0.83	0.82	0.71	0.44
	open	0.4	0.39	0.4	0.16	
C_{10}	closed	0.90	0.97	0.93	0.95	0.79
	open	0.80	0.78	0.79	0.63	
C_{11}	closed	0.97	0.97	0.97	0.96	0.73
	open	0.66	0.66	0.66	0.49	
C_{12}	closed	0.83	0.50	0.62	0.45	0.56
	open	0.73	0.91	0.81	0.66	
C_{13}	closed	0.96	0.91	0.93	0.9	0.91
	open	0.94	0.98	0.95	0.91	
C_{14}	closed	0.56	0.61	0.58	0.41	0.56
	open	0.85	0.82	0.83	0.70	
C_{15}	closed	0.72	0.74	0.73	0.62	0.59
	open	0.76	0.74	0.75	0.56	
C_{16}	closed	0.64	0.67	0.66	0.46	0.27
	open	0.27	0.26	0.26	0.07	
C_{17}	closed	0.85	0.95	0.90	0.88	0.88
	open	0.95	0.92	0.93	0.87	
C_{18}	closed	0.92	0.89	0.9	0.84	0.84
	open	0.91	0.93	0.92	0.84	
C_{19}	closed	0.75	0.76	0.76	0.63	0.32
	open	0.07	0.06	0.07	0.01	
C_{20}	closed	0.9	0.94	0.92	0.92	0.68
	open	0.66	0.63	0.64	0.43	
Avg.	closed	0.71	0.77	0.72	0.68	0.59
	open	0.63	0.62	0.62	0.50	

Table A.6 – Safety bar detection (by chairlift) results in *LOCO* setting, using Faster R-CNN model with VGG16 backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	closed	0.94	0.56	0.7	0.55	0.34
	open	0.26	0.22	0.24	0.12	
C_1	closed	0.97	0.97	0.97	0.95	0.80
	open	0.79	0.77	0.78	0.65	
C_2	closed	0.00	0.00	0.00	0.00	0.47
	open	0.98	0.97	0.98	0.95	
C_3	closed	1.00	1.00	1.00	1.00	0.98
	open	0.98	0.98	0.98	0.97	
C_4	closed	0.96	0.96	0.96	0.93	0.88
	open	0.91	0.89	0.9	0.83	
C_5	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_6	closed	1.00	1.00	1.00	1.00	1.00
	open	1.00	1.00	1.00	1.00	
C_7	closed	0.72	0.72	0.72	0.55	0.28
	open	0.00	0.00	0.00	0.00	
C_8	closed	0.99	0.99	0.99	0.99	0.88
	open	0.99	0.77	0.86	0.76	
C_9	closed	1.00	1.00	1.00	1.00	0.99
	open	1.00	0.99	0.99	0.99	
C_{10}	closed	0.99	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{11}	closed	0.99	0.99	0.99	0.98	0.99
	open	1.00	1.00	1.00	1.00	
C_{12}	closed	0.67	0.1	0.13	0.09	0.54
	open	0.99	0.99	0.99	0.99	
C_{13}	closed	1.00	0.89	0.93	0.89	0.95
	open	1.00	1.00	1.00	1.00	
C_{14}	closed	0.99	0.99	0.99	0.98	0.98
	open	1.00	0.99	0.99	0.99	
C_{15}	closed	0.99	0.98	0.98	0.98	0.98
	open	1.00	0.99	1.00	0.99	
C_{16}	closed	1.00	1.00	1.00	1.00	0.98
	open	1.00	0.97	0.98	0.97	
C_{17}	closed	0.98	0.98	0.98	0.97	0.98
	open	1.00	0.99	1.00	0.99	
C_{18}	closed	1.00	0.99	0.99	0.99	0.99
	open	1.00	1.00	1.00	1.00	
C_{19}	closed	0.99	0.99	0.99	0.98	0.79
	open	0.77	0.76	0.77	0.6	
C_{20}	closed	1.00	1.00	1.00	1.00	0.93
	open	0.93	0.92	0.92	0.86	
Avg.	closed	0.91	0.86	0.87	0.85	0.84
	open	0.89	0.87	0.88	0.84	

A.2 People detection

In section 3.4 we provided the implementation details of this experiment and average results over all the chairlifts. In this section, we show detailed results for each chairlift independently in the three settings *i.e.* *OOC*, *LOCO*, and *All*.

A.2.1 OOC

ZF Backbone

Faster R-CNN produce 0.76 mAP in *OOC* setting using both feature extractor ZF and VGG16, 0.54 for 'child' class and between 0.97 and 0.98 for 'adult' class. Which implies that the detector is able to detect with height precision and recall the 'adult' objects even if the dataset is small and with too few variations. However many 'child' objects are left behind without being detected which led to low recall which in its turn led to poor AP precision, even though the precision is not too bad. Table A.7 shows the results for each chairlift independently when ZF backbone is used.

VGG16 Backbone

Table A.8 shows the results for each chairlift independently when VGG16 backbone is used. The mAP achieved by Faster R-CNN when the very deep VGG16 model used as feature extractor wasn't noticeably increased.

A.2.2 All

ZF Backbone

In the contrary to safety bar detection, no better result was obtained in this setting, because people objects are not chairlift-dependent, so the enormous size of the dataset did not add extra variation or make the objects more well-represented. Instead Faster R-CNN with ZF backbone has lower performance than *OOC* setting. Table A.9 shows the results for each chairlift independently.

VGG16 Backbone

The mAP achieved by Faster R-CNN when VGG16 used as feature extractor increased to 0.76 (0.1 better than ZF feature extractor). Table A.10 shows the results for each chairlift independently. In this experiment in general it is worthy to use deeper network as backbone, but it is not better than using any backbone in *OOC* setting.

A.2.3 LOCO

ZF Backbone

In *LOCO* experiment with ZF backbone, the performance of 'child' classes decreased for all the measures, mAP for the different chairlifts has a range between 0 and 0.51 (see Table A.11).

VGG16 Backbone

Using VGG16 backbone in *LOCO* experiments reduce the harm in performance, with mAP equal to 0.7 (0.09 better than ZF backbone). The detector profited from the deeper and more expressive features of the VGG16. For 'adult' class the performance is equivalent to the other settings where test images belong to a chairlift used for training too, because as mentioned before people objects are not dependent on the chairlift. Table A.12 shows the results for each chairlift independently.

Table A.7 – People detection (by chairlift) results in *OOC* setting, using Faster R-CNN model with ZF backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	child	0.91	0.67	0.77	0.66	0.81
	adult	0.90	0.98	0.94	0.97	
C_1	child	0.85	0.73	0.79	0.69	0.83
	adult	0.94	0.99	0.96	0.98	
C_2	child	0.93	0.58	0.69	0.57	0.77
	adult	0.95	0.98	0.97	0.98	
C_3	child	0.92	0.78	0.84	0.77	0.88
	adult	0.93	0.99	0.96	0.98	
C_4	child	0.92	0.57	0.69	0.56	0.78
	adult	0.97	0.99	0.98	0.99	
C_5	child	0.83	0.70	0.75	0.66	0.81
	adult	0.92	0.97	0.95	0.96	
C_6	child	0.87	0.55	0.67	0.53	0.76
	adult	0.97	0.99	0.98	0.98	
C_7	child	0.83	0.36	0.48	0.32	0.66
	adult	0.93	1.00	0.96	0.99	
C_8	child	0.87	0.51	0.63	0.48	0.73
	adult	0.93	0.99	0.96	0.99	
C_9	child	0.87	0.40	0.54	0.37	0.68
	adult	0.97	0.98	0.97	0.98	
C_{10}	child	0.82	0.64	0.70	0.61	0.79
	adult	0.95	0.98	0.96	0.98	
C_{11}	child	0.94	0.54	0.65	0.52	0.76
	adult	0.97	0.99	0.98	0.99	
C_{12}	child	0.86	0.68	0.75	0.66	0.81
	adult	0.94	0.98	0.96	0.97	
C_{13}	child	0.88	0.70	0.78	0.69	0.81
	adult	0.95	0.93	0.94	0.93	
C_{14}	child	0.90	0.70	0.78	0.68	0.83
	adult	0.98	0.99	0.98	0.99	
C_{15}	child	0.89	0.69	0.77	0.67	0.82
	adult	0.90	0.99	0.94	0.97	
C_{17}	child	0.92	0.53	0.66	0.50	0.74
	adult	0.97	0.98	0.98	0.98	
C_{18}	child	0.84	0.67	0.74	0.63	0.80
	adult	0.94	0.99	0.96	0.98	
C_{19}	child	0.87	0.68	0.76	0.66	0.82
	adult	0.96	0.98	0.97	0.98	
C_{20}	child	0.88	0.53	0.65	0.50	0.74
	adult	0.93	0.99	0.96	0.98	
C_{21}	child	0.67	0.19	0.28	0.14	0.54
	adult	0.94	0.94	0.94	0.93	
C_{22}	child	0.78	0.12	0.19	0.11	0.54
	adult	0.97	0.97	0.97	0.96	
C_{23}	child	0.97	0.55	0.69	0.54	0.77
	adult	0.96	0.99	0.98	0.99	
Avg.	child	0.87	0.57	0.66	0.54	0.76
	adult	0.95	0.98	0.96	0.98	

Table A.8 – People detection (by chairlift) results in *OOC* setting, using Faster R-CNN model with VGG16 backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	child	0.92	0.69	0.79	0.68	0.82
	adult	0.92	0.98	0.95	0.97	
C_1	child	0.87	0.68	0.76	0.66	0.82
	adult	0.95	0.98	0.96	0.98	
C_2	child	0.96	0.42	0.57	0.42	0.70
	adult	0.94	0.99	0.97	0.99	
C_3	child	0.94	0.79	0.86	0.78	0.88
	adult	0.94	0.99	0.96	0.98	
C_4	child	0.94	0.52	0.66	0.51	0.75
	adult	0.97	0.99	0.98	0.99	
C_5	child	0.87	0.69	0.77	0.67	0.81
	adult	0.94	0.97	0.95	0.96	
C_6	child	0.89	0.54	0.67	0.52	0.75
	adult	0.97	0.99	0.98	0.98	
C_7	child	0.96	0.39	0.54	0.38	0.69
	adult	0.95	0.99	0.97	0.99	
C_8	child	0.94	0.55	0.69	0.53	0.76
	adult	0.95	0.99	0.97	0.99	
C_9	child	0.81	0.45	0.56	0.42	0.70
	adult	0.97	0.98	0.97	0.98	
C_{10}	child	0.90	0.63	0.73	0.60	0.79
	adult	0.95	0.98	0.97	0.98	
C_{11}	child	0.94	0.37	0.52	0.37	0.68
	adult	0.97	0.99	0.98	0.99	
C_{12}	child	0.91	0.68	0.78	0.66	0.81
	adult	0.95	0.98	0.96	0.97	
C_{13}	child	0.91	0.75	0.82	0.75	0.84
	adult	0.96	0.94	0.95	0.93	
C_{14}	child	0.92	0.69	0.79	0.68	0.83
	adult	0.98	1.00	0.99	0.99	
C_{15}	child	0.91	0.66	0.76	0.65	0.81
	adult	0.93	0.98	0.95	0.97	
C_{17}	child	0.98	0.50	0.64	0.49	0.73
	adult	0.97	0.98	0.97	0.97	
C_{18}	child	0.88	0.64	0.74	0.62	0.80
	adult	0.94	0.99	0.97	0.98	
C_{19}	child	0.92	0.63	0.75	0.62	0.79
	adult	0.96	0.98	0.97	0.97	
C_{20}	child	0.91	0.56	0.68	0.55	0.77
	adult	0.95	0.99	0.97	0.98	
C_{21}	child	0.74	0.32	0.44	0.30	0.61
	adult	0.95	0.94	0.94	0.93	
C_{22}	child	0.98	0.14	0.23	0.13	0.54
	adult	0.97	0.97	0.97	0.96	
C_{23}	child	0.98	0.43	0.59	0.42	0.70
	adult	0.96	0.99	0.98	0.99	
Avg.	child	0.91	0.55	0.67	0.54	0.76
	adult	0.95	0.98	0.97	0.97	

Table A.9 – People detection (by chairlift) results in the setting *All*, using Faster R-CNN model with ZF backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	child	0.88	0.42	0.55	0.39	0.68
	adult	0.83	0.98	0.90	0.96	
C_1	child	0.91	0.42	0.55	0.40	0.70
	adult	0.90	1.00	0.94	0.99	
C_2	child	0.98	0.40	0.54	0.40	0.70
	adult	0.93	0.99	0.96	0.99	
C_3	child	0.91	0.32	0.46	0.31	0.64
	adult	0.81	0.99	0.89	0.97	
C_4	child	0.96	0.31	0.41	0.29	0.64
	adult	0.90	1.00	0.94	0.99	
C_5	child	0.91	0.39	0.54	0.37	0.67
	adult	0.85	0.97	0.91	0.96	
C_6	child	0.98	0.34	0.49	0.34	0.67
	adult	0.95	0.99	0.97	0.99	
C_7	child	0.77	0.14	0.23	0.14	0.56
	adult	0.85	1.00	0.92	0.99	
C_8	child	0.89	0.39	0.52	0.37	0.68
	adult	0.86	0.99	0.92	0.98	
C_9	child	0.86	0.22	0.34	0.21	0.59
	adult	0.96	0.98	0.97	0.97	
C_{10}	child	0.86	0.46	0.58	0.44	0.69
	adult	0.90	0.96	0.93	0.95	
C_{11}	child	1.00	0.25	0.37	0.25	0.62
	adult	0.96	0.99	0.98	0.99	
C_{12}	child	0.95	0.37	0.53	0.36	0.67
	adult	0.91	0.99	0.95	0.98	
C_{13}	child	0.88	0.48	0.61	0.46	0.69
	adult	0.90	0.93	0.91	0.92	
C_{14}	child	0.96	0.45	0.61	0.44	0.71
	adult	0.95	1.00	0.97	0.99	
C_{15}	child	0.86	0.51	0.63	0.48	0.71
	adult	0.83	0.97	0.89	0.94	
C_{17}	child	0.92	0.16	0.26	0.15	0.56
	adult	0.93	0.99	0.96	0.98	
C_{18}	child	0.85	0.47	0.59	0.45	0.71
	adult	0.88	0.99	0.93	0.97	
C_{19}	child	0.91	0.36	0.51	0.35	0.67
	adult	0.91	0.99	0.95	0.98	
C_{20}	child	0.93	0.17	0.28	0.17	0.57
	adult	0.89	1.00	0.94	0.98	
C_{21}	child	0.88	0.12	0.20	0.11	0.50
	adult	0.86	0.92	0.89	0.89	
C_{22}	child	0.69	0.15	0.23	0.14	0.54
	adult	0.93	0.96	0.94	0.95	
C_{23}	child	1.00	0.22	0.35	0.22	0.60
	adult	0.95	1.00	0.97	0.99	
Avg.	child	0.90	0.36	0.50	0.34	0.66
	adult	0.90	0.98	0.94	0.97	

Table A.10 – People detection (by chairlift) results in the setting *All*, using Faster R-CNN model with VGG16 backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	child	0.87	0.77	0.82	0.73	0.85
	adult	0.90	0.98	0.94	0.97	
C_1	child	0.90	0.62	0.72	0.60	0.79
	adult	0.92	0.99	0.96	0.99	
C_2	child	0.99	0.58	0.73	0.58	0.78
	adult	0.95	0.99	0.97	0.98	
C_3	child	0.91	0.59	0.71	0.57	0.78
	adult	0.87	1.00	0.93	0.99	
C_4	child	0.95	0.55	0.69	0.55	0.77
	adult	0.95	0.99	0.97	0.99	
C_5	child	0.90	0.62	0.73	0.61	0.79
	adult	0.90	0.99	0.94	0.98	
C_6	child	0.99	0.49	0.64	0.49	0.74
	adult	0.96	1.00	0.98	0.99	
C_7	child	1.00	0.49	0.65	0.49	0.74
	adult	0.93	1.00	0.96	0.99	
C_8	child	0.97	0.65	0.77	0.64	0.81
	adult	0.93	0.99	0.96	0.99	
C_9	child	0.90	0.33	0.45	0.31	0.65
	adult	0.97	0.99	0.98	0.98	
C_{10}	child	0.86	0.65	0.72	0.62	0.79
	adult	0.94	0.98	0.96	0.97	
C_{11}	child	1.00	0.38	0.53	0.38	0.69
	adult	0.97	1.00	0.98	1.00	
C_{12}	child	0.92	0.62	0.74	0.60	0.79
	adult	0.93	0.99	0.96	0.98	
C_{13}	child	0.89	0.65	0.74	0.62	0.78
	adult	0.92	0.94	0.93	0.94	
C_{14}	child	0.97	0.58	0.72	0.57	0.78
	adult	0.96	1.00	0.98	0.99	
C_{15}	child	0.88	0.65	0.74	0.63	0.79
	adult	0.90	0.98	0.94	0.96	
C_{17}	child	0.94	0.38	0.53	0.38	0.68
	adult	0.95	0.99	0.97	0.98	
C_{18}	child	0.93	0.59	0.70	0.57	0.78
	adult	0.93	1.00	0.96	0.99	
C_{19}	child	0.96	0.53	0.68	0.52	0.75
	adult	0.93	0.99	0.96	0.98	
C_{20}	child	0.97	0.43	0.59	0.42	0.70
	adult	0.92	1.00	0.95	0.99	
C_{21}	child	0.88	0.28	0.41	0.27	0.60
	adult	0.93	0.96	0.94	0.94	
C_{22}	child	0.88	0.20	0.32	0.18	0.57
	adult	0.97	0.97	0.97	0.96	
C_{23}	child	1.00	0.40	0.56	0.40	0.70
	adult	0.96	0.99	0.97	0.99	
Avg.	child	0.91	0.56	0.69	0.54	0.76
	adult	0.93	0.99	0.96	0.98	

Table A.11 – People detection (by chairlift) results in *LOCO* setting, using Faster R-CNN model with ZF backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	child	0.86	0.13	0.22	0.12	0.54
	adult	0.82	0.99	0.89	0.96	
C_1	child	0.69	0.36	0.45	0.33	0.66
	adult	0.86	0.99	0.92	0.98	
C_2	child	0.98	0.51	0.66	0.51	0.74
	adult	0.94	0.98	0.96	0.97	
C_3	child	0.89	0.21	0.33	0.20	0.58
	adult	0.81	0.99	0.89	0.97	
C_4	child	0.91	0.35	0.50	0.34	0.66
	adult	0.90	0.99	0.94	0.98	
C_5	child	0.93	0.24	0.37	0.23	0.59
	adult	0.88	0.98	0.93	0.96	
C_6	child	0.96	0.24	0.38	0.24	0.61
	adult	0.95	0.99	0.97	0.99	
C_7	child	0.74	0.37	0.49	0.31	0.65
	adult	0.78	0.99	0.87	0.98	
C_8	child	0.88	0.41	0.53	0.38	0.67
	adult	0.81	0.98	0.89	0.96	
C_9	child	0.86	0.23	0.36	0.22	0.59
	adult	0.96	0.97	0.97	0.97	
C_{10}	child	0.77	0.53	0.63	0.48	0.70
	adult	0.91	0.95	0.93	0.92	
C_{11}	child	0.97	0.32	0.47	0.31	0.65
	adult	0.97	0.99	0.98	0.99	
C_{12}	child	0.91	0.19	0.31	0.18	0.57
	adult	0.93	0.97	0.95	0.96	
C_{13}	child	0.90	0.25	0.38	0.23	0.57
	adult	0.91	0.92	0.91	0.90	
C_{14}	child	0.98	0.29	0.44	0.29	0.64
	adult	0.95	1.00	0.97	0.99	
C_{15}	child	0.76	0.55	0.62	0.48	0.69
	adult	0.85	0.93	0.88	0.90	
C_{17}	child	0.86	0.13	0.23	0.12	0.55
	adult	0.93	0.99	0.96	0.98	
C_{18}	child	0.81	0.52	0.62	0.49	0.72
	adult	0.91	0.97	0.94	0.95	
C_{19}	child	0.84	0.33	0.47	0.30	0.64
	adult	0.90	0.99	0.94	0.98	
C_{20}	child	0.93	0.15	0.25	0.14	0.56
	adult	0.90	0.99	0.94	0.98	
C_{21}	child	0.71	0.26	0.36	0.22	0.53
	adult	0.86	0.88	0.87	0.84	
C_{22}	child	0.00	0.00	0.00	0.00	0.20
	adult	0.98	0.40	0.56	0.39	
C_{23}	child	0.93	0.45	0.61	0.45	0.71
	adult	0.96	0.99	0.97	0.98	
Avg..	child	0.83	0.31	0.42	0.29	0.61
	adult	0.90	0.95	0.91	0.93	

Table A.12 – People detection (by chairlift) results in *LOCO* setting, using Faster R-CNN model with VGG16 backbone pretrained on ImageNet.

chairlift	class	prec	rec	F1	AP	mAP
C_0	child	0.91	0.34	0.47	0.32	0.65
	adult	0.83	0.99	0.90	0.98	
C_1	child	0.91	0.55	0.67	0.53	0.76
	adult	0.91	0.99	0.95	0.98	
C_2	child	0.98	0.61	0.74	0.60	0.79
	adult	0.94	0.98	0.96	0.98	
C_3	child	0.94	0.50	0.65	0.48	0.73
	adult	0.85	1.00	0.92	0.99	
C_4	child	0.97	0.46	0.62	0.45	0.72
	adult	0.93	0.99	0.96	0.99	
C_5	child	0.95	0.36	0.50	0.36	0.67
	adult	0.89	0.99	0.94	0.98	
C_6	child	0.96	0.37	0.52	0.36	0.68
	adult	0.96	0.99	0.97	0.99	
C_7	child	0.77	0.49	0.59	0.43	0.70
	adult	0.80	0.99	0.88	0.98	
C_8	child	0.93	0.43	0.56	0.42	0.70
	adult	0.89	0.99	0.94	0.99	
C_9	child	0.90	0.38	0.53	0.36	0.67
	adult	0.97	0.98	0.97	0.98	
C_{10}	child	0.81	0.66	0.72	0.61	0.78
	adult	0.95	0.97	0.96	0.96	
C_{11}	child	1.00	0.42	0.58	0.41	0.70
	adult	0.97	1.00	0.98	1.00	
C_{12}	child	0.95	0.35	0.50	0.34	0.66
	adult	0.92	0.99	0.95	0.98	
C_{13}	child	0.93	0.42	0.56	0.40	0.67
	adult	0.91	0.94	0.92	0.93	
C_{14}	child	0.96	0.46	0.62	0.45	0.72
	adult	0.96	0.99	0.98	0.99	
C_{15}	child	0.77	0.75	0.75	0.70	0.81
	adult	0.91	0.94	0.92	0.92	
C_{17}	child	0.90	0.40	0.54	0.37	0.68
	adult	0.95	1.00	0.97	0.99	
C_{18}	child	0.89	0.62	0.72	0.60	0.79
	adult	0.93	0.99	0.96	0.98	
C_{19}	child	0.93	0.55	0.69	0.53	0.76
	adult	0.93	0.99	0.96	0.98	
C_{20}	child	0.98	0.28	0.44	0.28	0.63
	adult	0.91	0.99	0.95	0.98	
C_{21}	child	0.79	0.34	0.43	0.29	0.58
	adult	0.92	0.89	0.91	0.87	
C_{22}	child	0.92	0.17	0.28	0.16	0.56
	adult	0.95	0.97	0.96	0.96	
C_{23}	child	0.97	0.54	0.69	0.53	0.76
	adult	0.96	0.99	0.98	0.99	
Avg..	child	0.91	0.45	0.58	0.43	0.70
	adult	0.92	0.98	0.95	0.97	

Appendix B

French translations

Table des matières

B.1 Introduction

Les stations de ski sont une des destinations touristiques préférées en hiver dans presque tous les pays qui ont des régions de montagne et un hiver enneigé. La France est la destination de ski la plus populaire en Europe; elle possède le plus grand nombre de télésièges au monde (plus de 3000). Chaque année, parmi les millions de voyages en télésiège, seuls quelques accidents surviennent (28 accidents sur les 403 millions de voyages en télésiège en France pour la saison 2018/2019, selon le rapport sur les accidents des remontées mécaniques en France et concernant les usagers transportés pendant la saison 2018/2019 en [4]), mais ils sont potentiellement graves et parfois mortels. De nombreux facteurs influencent le risque d'accident, notamment les conditions météorologiques, la circulation sur les pistes, l'expérience et les compétences des passagers, *etc*. Néanmoins, la plupart des accidents sont causés par l'erreur humaine et les comportements à risque. Les télésièges sont équipés d'une barre de sécurité (ou garde-corps) pour maintenir les passagers dans leurs sièges de'une manière sûre. C'est aux passagers de soulever et d'abaisser la barre de sécurité lorsqu'ils montent ou descendent du véhicule (voir Fig 1). Toutefois, les stations disposent d'un système de contrôle qui surveille les télésièges à l'embarquement pour assurer que les passagers ont correctement fermé la barre de sécurité, et au cas où ils ne l'auraient pas fait, ils sont avertis par le système. Ce système de contrôle est généralement composé d'opérateurs humains qui surveillent et détectent toute situation dangereuse à l'embarquement et à l'arrivée.

Pour renforcer la sécurité sur les télésièges, la start-up française Bluecime a développé un système qui s'appelle "Système Intelligent de Vision Artificielle par Ordinateur" (Sivao). Le système proposé est un système de vision par ordinateur dont l'objectif est d'aider les opérateurs de remontées mécaniques à détecter les anomalies lors de l'embarquement sur les télésièges, et d'avertir les utilisateurs en cas de situations dangereuses et prévenir les accidents éventuels. Le système se compose d'une caméra, d'un ordinateur et d'une alarme. La caméra enregistre la scène d'embarquement. Ensuite, certaines images des vidéos sont traitées en temps réel à l'aide de techniques de traitement d'images non basées sur l'apprentissage, pour détecter: (i) l'occupation du télésiège, et (ii) la position de la barre de sécurité. Lorsqu'une situation dangereuse est détectée *i.e*. le télésiège est occupé, et la barre de sécurité n'est pas complètement fermée, l'alarme se déclenche.

Bien que les décisions prises par Sivao soient très précises, l'installation pour un nouveau télésiège nécessite des configurations manuelles longues et coûteuses. Étant donné qu'un nombre croissant de télésièges sont équipés de ce système, la configuration de l'installation nécessite un temps et des efforts considérables. De ce fait, dans le cadre du projet "Montagne Innovante, Vision et Apprentissage par Ordinateur" (Mivao), l'idée est d'utiliser l'apprentissage profond pour améliorer les performances de Sivao et pour réduire les ajustements manuels coûteux dans des différents environnements et circonstances.

Les dernières avancées en matière de modèles basés sur l'apprentissage profond ont apporté des solutions originales qui attirent l'attention de l'industrie. Notamment, les approches basées sur les réseaux neuronaux convolutifs profonds (CNN), qui ont montré des améliorations substantielles dans la généralisation d'un large ensemble de problèmes, allant de simples tâches de reconnaissance d'images, telles que la reconnaissance de chiffres manuscrits (début des CNN), à des tâches plus complexes telles que la prédiction de l'impact de la modification de séquences d'ADN non codantes sur l'expression des gènes. L'avantage principal des méthodes d'apprentissage profond est qu'elles permettent d'apprendre automatiquement et progressivement des caractéristiques de haut niveau à partir des données, en utilisant une procédure d'apprentissage général. Les méthodes d'apprentissage profond sont alimentées par une énorme quantité de données, alors qu'elles ne nécessitent qu'un pré-traitement manuel nominal et une ingénierie de caractéristiques négligeable, ce qui annule la nécessité d'une expertise dans le domaine, contrairement aux techniques d'apprentissage automatique traditionnelles.

Cette thèse fait partie du projet Mivao visant à sécuriser l'embarquement sur les télésièges. Nous abordons ce problème en proposant des techniques d'apprentissage profond pour améliorer Sivao, et exploiter les données disponibles.

Axes de recherche

Mivao vise à créer un système intelligent pour effectuer une analyse intensive de la scène d'embarquement et identifier les situations rares mais potentiellement dangereuses dans les

télésièges. Ayant les fonctionnalités de SIVAO (c'est-à-dire la détection en temps réel de l'occupation du télésiège et de la position de la barre de sécurité), MIVAO devrait être plus fiable et moins sensible aux conditions changeantes, compte tenu de la réduction de la charge de travail de configuration lors de l'installation du système dans de nouveaux sites. Presque tous les télésièges sont surveillés par des opérateurs humains au poste d'embarquement; le travail de ces opérateurs consiste à surveiller les situations d'anomalie et à y réagir. Cependant, la surveillance manuelle continue est laborieuse pour les humains, l'analyse automatique des vidéos de surveillance est donc une solution à ce type de tâches fastidieuses. L'analyse des vidéos de surveillance se fait principalement à deux niveaux: (i) le premier niveau, où les éléments essentiels sont détectés, et (ii) le second niveau, où les résultats du premier niveau sont utilisés pour la prise de décision. L'analyse de premier niveau implique des différents algorithmes d'analyse d'images et de vidéos provenant de domaines de recherche tels que la classification, la détection d'objets, le suivi d'objets, la reconnaissance d'actions, *etc*. Dans le problème de la sécurité des télésièges, les éléments essentiels sont les personnes (il est indispensable de différencier les adultes des enfants) et la barre de sécurité (il est impératif de déterminer sa position: complètement ouverte, complètement fermée, ou entre les deux). C'est pour cela que -dans cette thèse- nous nous concentrons sur la détection et la classification adulte/enfant et barre de sécurité ouverte/fermée. Pour répondre aux besoins de MIVAO, nous tentons de résoudre le problème par l'exploitation des données disponibles pour former des modèles d'apprentissage profond robustes avec le minimum d'annotations possible, et cela en tenant compte des contraintes géométriques afin d'améliorer la détection. En conséquence, nous choisissons d'étudier la détection adaptative d'objets dans le domaine non supervisé et la classification d'images.

Tout d'abord, il est pertinent de clarifier la confusion occasionnelle entre la tâche de classification d'images et celle de détection d'objets: la *classification d'images* fait référence à la tâche de nommer l'objet dominant dans une image, par contre, la *détection d'objets* consiste à nommer et à localiser tous les objets dans l'image. L'apprentissage supervisé consiste à former le système en le récompensant s'il donne le résultat attendu pour une entrée donnée et en le pénalisant s'il ne le fait pas. Les classificateurs d'images sont destinés à produire la classe de l'image; par conséquent, ils sont formés en utilisant des annotations au niveau de l'image. D'autre part, les détecteurs d'objets nécessitent des annotations de type boîte englobante autour de chaque objet dans les images de formation. Ces annotations sont essentielles pour l'apprentissage supervisé, mais en même temps, elles sont coûteuses et prennent beaucoup de temps car elles sont généralement générées manuellement.

La prédiction de la sortie pour les nouvelles entrées sur lesquelles le système n'a pas été formé est appelée *généralisation*. Si les données de formation sont représentatives de la distribution correspondante, alors le système se généralise bien. Cependant, si les entrées au moment du test sont significativement incohérentes avec les données de formation, le modèle peut ne pas bien se généraliser. Pour cette raison, la généralisation vers des distributions plus larges semble aller trop loin, alors que le ciblage d'une distribution particulière est plus réalisable. Cet problématique est appelé *l'adaptation de domaine*. Les données de formation sont utilisées comme source d'informations supplémentaires pour le domaine cible. Si aucune étiquette n'est disponible dans le domaine cible, on parle de *l'adaptation de domaine non supervisée*. Le défi consiste à surmonter la différence entre les domaines, dans la mesure où un système formé dans le domaine source se généralisera bien au domaine cible.

Le cœur de cette thèse se concentre sur le défi de la généralisation à des nouveaux domaines avec un minimum d'annotations en exploitant les contraintes géométriques. Dans ce problème, chaque télésiège est considéré comme un nouveau domaine, en raison des variations importantes entre les télésièges en termes de forme, de taille, de nombre de sièges, d'orientation, *etc*. (voir la Fig 4 pour des exemples).

Motivation et contributions

Dans le cadre du projet MIVAO, une grande base d'images était disponible sans annotations. Notre objectif était de trouver un moyen efficace d'utiliser ces données avec le moins d'annotations possible. Dans cette thèse, nous concentrons sur l'adaptation de domaine non supervisée pour la détection d'objets et la classification d'images. Nous proposons des solutions pour améliorer le processus d'apprentissage et donc la précision, pour généraliser et atteindre des performances compétitives là où peu ou pas de données annotées sont disponibles.

Une première contribution de cette thèse a été d'appliquer une méthode de détection d'objets basée sur CNN à l'ensemble des données des télésièges afin d'évaluer sa capacité à détecter la barre de sécurité et les personnes dans l'image, afin d'identifier éventuellement les situations à risque. Comme prévu, cette approche donne de meilleurs résultats qu'un classificateur classique formé sans utiliser des annotations de boîtes englobantes. Cependant, la formation de ces modèles nécessite un grand nombre d'annotations, mais l'un des objectifs les plus importants de ce projet est de trouver une solution qui s'adapte automatiquement lorsqu'elle est appliquée aux nouvelles données non annotées. Par conséquent, **l'adaptation de domaine pour la détection d'objets** est le premier axe étudié dans cette thèse. Nous nous concentrons sur la question suivante: quelles caractéristiques doivent être adaptées dans un détecteur d'objets pour sa généralisation à partir d'un domaine source vers un domaine cible? Peu de travaux abordent explicitement le problème de l'adaptation de domaine non supervisée pour la détection d'objets. Les approches existantes ont ajouté des éléments de formation contradictoires dans le détecteur classique Faster R-CNN, tant au niveau global qu'au niveau des instances sans adapter le réseau de proposition de région (RPN), ce qui a entraîné un déplacement de domaine résiduel. Nous avons proposé d' **adapter le RPN** pour garantir que les caractéristiques extraites des images cibles chevauchent les caractéristiques de l'objet source.

Cependant, même le fait de n'annoter que le domaine source est très coûteux lorsqu'il s'agit de délimiter des boîtes au niveau de l'instance. En outre, l'adaptation de domaine n'est pas toujours satisfaisante à l'échelle de dizaines de domaines cibles différents; le facteur de réussite le plus critique de l'adaptation de domaine est le niveau de similitude entre le domaine source et le domaine cible. Pour la deuxième contribution concernant les coûts liés aux annotations des boîtes englobantes, nous avons opté pour **l'utilisation de masques binaires pour guider un classificateur**. Nous nous concentrons sur la question suivante: comment exploiter les contraintes géométriques comme information a priori pour concevoir un classificateur d'images et pour le former sur seulement quelques images annotées? L'idée principale est d'informer le classificateur de l'élément sur lequel il doit se concentrer dans l'image pour prendre sa décision, en utilisant un masque binaire, ce qui élimine le besoin d'annotations de boîte englobante qui sont beaucoup plus coûteuses. En utilisant une architecture siamoise, nous alimentons un classificateur d'images basé sur CNN avec l'image et le masque binaire, et nous forçons le classificateur à n'extraire que les caractéristiques souhaitées en fonction du masque binaire.

Lorsque la tentative a été couronnée de succès, la question était: ce classificateur est-il capable de se généraliser? De plus, comment améliorer sa généralisabilité? Nous proposons une étape de mise au point qui n'utilise que deux masques du domaine cible et qui donne des résultats exceptionnels sans nécessiter de composante d'adaptation de domaine spécifique. En outre, nous montrons que des masques virtuels spécifiques renforcent la propriété de généralisation du réseau pour les nouveaux télésièges jamais vus sans nécessiter d'images ni de masques de ces télésièges cibles. Cette dernière approche est également plus performante que la détection d'objets, et elle n'a pas besoin d'annotations au niveau de la boîte englobante par exemple.

Comme mentionné ci-dessus, le cas d'utilisation principal considéré dans cette thèse est le problème de la sécurité des télésièges. Bien que la motivation de notre travail soit d'aborder les problèmes de Mivao, les contributions apportées ne se limitent pas à ce contexte, mais pourraient être utilisées dans toute application nécessitant une analyse de la scène, et pourraient être utiles dans plusieurs scénarios tels que la conduite autonome, comme nous le montrerons dans nos résultats expérimentaux.

B.1.1 Plan

Cette thèse vise à proposer des méthodes basées sur l'apprentissage profond pour identifier les situations dangereuses dans les télésièges. Ces méthodes sont destinées à améliorer les performances du système Sivao existant en améliorant la précision et en réduisant le temps nécessaire à la configuration du système pour un nouveau télésiège. Les chapitres de ce manuscrit sont organisés comme suit:

- Chapitre 1 présente le projet Mivao, ses objectifs, ses défis et ses perspectives, et fournit une description détaillée de son ensemble de données. De plus, il présente nos objectifs liés à Mivao dans le cadre de cette thèse, faisant partie du projet.
- Chapitre 2 présente un survol de l'état de l'art des aspects méthodologiques liés à nos

objectifs. Dans ce chapitre, nous passons en revue les solutions qui existent déjà dans le domaine de la sécurité des télésièges. Nous étudions également les méthodes de classification d'images basées sur l'apprentissage profond. Ensuite, nous présentons la détection d'objets et discutons de ses principaux avantages pour le problème de la sécurité des télésièges. L'aspect suivant à considérer est l'adaptation au domaine. Enfin, nous passons en revue ce qui a déjà été proposé pour exploiter une connaissance a priori particulière dans des modèles d'apprentissage profond.

- Chapitre 3 comprend une contribution préliminaire: l'application de la détection d'objets au problème de la sécurité des télésièges. Ce chapitre évalue la performance de l'approche Faster R-CNN sur cette tâche et en souligne les principales limites.

- Chapitre 4 présente un nouveau point de vue sur le problème du décalage de domaine dans la détection d'objets. Nous proposons d'adapter le réseau de proposition de région (RPN) et d'intégrer ce nouveau module d'adaptation dans Faster R-CNN. En plus du problème de la sécurité des télésièges, nous menons des expériences dans le contexte de la conduite autonome pour comparer l'approche que nous proposons avec une autre méthode d'adaptation de domaine non supervisée appliquée à Faster R-CNN.

- Chapitre 5 présente une solution pour la tâche de classification des images basée sur CNN où la classe de chaque image dépend d'un petit détail de celle-ci. Nous proposons d'utiliser les réseaux siamois pour résoudre le problème de classification des images en utilisant des contraintes géométriques comme information a priori.

- Chapitre 6 a pour objectif de présenter une solution pour améliorer le processus d'apprentissage d'un réseau de classification lorsque moins d'images étiquetées sont nécessaires. Nous étudions le pouvoir de généralisation d'une telle approche, et nous proposons une solution pour l'améliorer.

- Finalement, en conclusion, nous synthétisons les travaux présentés dans les chapitres précédents. Ensuite, nous discutons de certaines perspectives de travail et des orientations de recherche possibles dans l'avenir.

B.2 Résumés des chapitres

B.2.1 Chapitre 1

Le projet Mivao a été lancé pour répondre aux besoins des opérateurs de stations de ski en matière de sécurisation des télésièges. Son objectif principal est de développer un système d'apprentissage et de vision par ordinateur pour faire une analyse de la scène d'embarquement, en temps réel. Mivao vise à développer un système permettant une analyse approfondie de la scène d'embarquement et de détecter non seulement les utilisateurs qui n'ont pas correctement fermé la barre de sécurité, mais aussi des situations plus sophistiquées comme un utilisateur qui a glissé de son véhicule, ou la présence d'enfants non accompagnés, *etc.* Les objectifs à atteindre pour la fiabilité de la détection des situations dangereuses sont les suivants:

- la fiabilité par rapport à la variabilité de la forme des télésièges;

- la fiabilité par rapport aux variations des conditions météorologiques telles que la pluie, la neige, le brouillard, *etc.* (voir la Fig 1.2 pour quelques exemples);

- la fiabilité par rapport aux variations des conditions d'éclairage pendant la journée (voir la Fig 1.3 pour quelques exemples);

- la fiabilité par rapport à la variabilité des tenues et des accessoires des utilisateurs, tels que les casques, les bonnets, les lunettes de ski, *etc.*;

- l'automatisation de la phase de configuration du système pour minimiser les coûts d'installation;

- la maîtrise des contraintes en temps réel, même si des algorithmes robustes sont utilisés.

Les méthodes d'apprentissage automatique et, en particulier, celles basées sur l'apprentissage profond, permettraient de déterminer plus précisément le risque lié à une situation donnée sur un télésiège et seraient en mesure de répondre de manière satisfaisante aux défis du projet.

L'ensemble des données du projet ne cesse d'augmenter. Au cours de cette thèse, nous avons utilisé des sous-ensembles de deux versions: 2018 et 2019. Il est à noter que la version 2018 est composée d'images provenant de 21 télésièges installés dans des stations de ski différentes. Pour des raisons de confidentialité, nous ne donnons pas les noms réels des télésièges, nous les appelons ci-après C_0, C_1, C_2, C_3, ..., C_{20}. Différentes classes globales sont fournies par Bluecime (*e.g.* fermé, ouvert, intermédiaire), ainsi que deux masques de segmentation binaire concernant la position (ouverte/fermée) de la barre de sécurité pour chaque télésiège. Nous avons proposé d'enrichir les annotations existantes et produire automatiquement des boîtes englobantes de la barre de sécurité. La localisation des différentes personnes (enfant/adulte) a été produite par des méthodes de "crowd sourcing". Nous précisons les tâches de reconnaissance principalement étudiées, *i.e.* la reconnaissance de la position de la barre de sécurité, et la détection des personnes.

B.2.2 Chapitre 2

Ce chapitre présente un survol de l'état de l'art des principaux aspects liés à nos objectifs. Tout d'abord, nous discutons des solutions qui existent déjà dans le domaine de la sécurité des télésièges. Nous présentons les approches générales de vidéo surveillance, et détaillons les initiatives récentes pour déployer des solutions d'intelligence artificielle pour l'analyse de situations anormales.

Deuxièmement, nous étudions les méthodes de classification d'images basées sur l'apprentissage profond. Nous décrivons les blocs élémentaires de ces modèles. Nous détaillons ensuite les réseaux de neurones convolutifs ainsi que leur entraînement par rétro-propagation du gradient de l'erreur, et la possibilité d'utiliser des réseaux pré-entraîné sur des données massivement annotées (transfer learning). Nous décrivons également les architectures convolutives modernes standardes, *e.g.* ResNet. Nous justifions la limitation des approches de classification dans notre contexte, notamment liée au manque d'information locale et à la difficulté d'interpréter les décisions.

Troisièmement, nous présentons la détection d'objets et essayons d'expliquer ses principaux avantages pour le problème de la sécurité des télésièges. Nous distinguons les approches effectuant la localisation en une seule étape (*e.g.* YOLO), et les méthodes basées sur deux étapes utilisant une étape de pré-sélection de régions (*e.g.* R-CNN). Ensuite, nous détaillons les évolutions des méthodes au cours des cinq dernières années.

Le quatrième aspect important à considérer est l'adaptation de domaine, c'est-à-dire la capacité de transférer un modèle précédemment appris d'un domaine source à un domaine cible, qui dans notre cas peut être un nouveau télésiège. Nous abordons les travaux autour de cette problématique, comme la généralisation de domaine, l'apprentissage avec peu (ou pas) d'exemples, et l'apprentissage multitâches.

Enfin, nous conclutons ce chapitre en présentant des méthodes qui peuvent être utilisées pour incorporer de l'information a priori et our améliorer les performances des modèles utilisant uniquement des données. Nous discutons de différentes possibilités pour cela, en particulier les modèles attentionnels, les modèles conditionnels, et les réseaux siamois.

B.2.3 Chapitre 3

Ce chapitre présente une étude expérimentale consistant à évaluer la faisabilité de détection automatique d'objets dans le problème du sécurité des télésièges. Nous utilisons un détecteur d'objets classique Faster R-CNN [3] pour détecter les éléments essentiels dans la scène *i.e.* la barre de sécurité (ouverte/fermée), et les personnes (adulte/enfant). Le détecteur d'objets est une solution pour guider le réseau vers les zones cruciales de l'image. Ensuite, au moment du test, nous pouvons l'utiliser comme modèle de classification en ignorant l'emplacement des boîtes englobantes.

Nos résultats montrent globalement la bonne performance de la méthode lorsque l'évaluation est effectuée sans décalage de domaine, et une dégradation forte des résultats sinon. D'un autre côté, apprendre à localiser correctement l'objet dans l'image est très prometteur pour améliorer

la précision de la classification. Cependant, cela nécessite des annotations coûteuses au niveau des instances.

B.2.4 Chapitre 4

Dans ce chapitre, nous présentons notre contribution concernant l'adaptation de domaine pour la détection d'objets. Nous nous appuyons sur le détecteur Faster R-CNN [3], et la méthode DA-Faster [106]. Nous menons des expériences approfondies dans deux contextes d'application différents: la conduite autonome et le problème de la sécurité des télésièges. Ce travail a été présenté dans un article [37] dans le cadre de *International Conference on Advanced Concepts for Intelligent Vision Systems, 2020.*

Après une analyse du flux de travail complet du détecteur Faster R-CNN classique, nous proposons d'adapter les caractéristiques tirées de ce réseau à deux niveaux différents: au niveau global dans le RPN et au niveau local pour chaque boîte englobante renvoyée par le RPN. Nous montrons que ces deux adaptations sont complémentaires et donnent de très bons résultats de détection. Pour l'adaptation de domaine, nous nous basons sur une fonction de coût adversaire classique.

B.2.5 Chapitre 5

Dans ce chapitre, nous proposons une solution originale pour exploiter la structure géométrique des images de télésiège, sans la nécessité d'annotation en forme de boîte englobante. Notre solution pourrait être appliquée aux tâches de classification d'images basées sur CNN, où la classe de chaque image dépend des petits détails qu'elle contient. L'idée principale est d'informer le classificateur des éléments de l'image sur lesquels il doit se concentrer pour prendre sa décision. Pour cela, nous utilisons une architecture siamoise et alimenté le réseau avec une image et un masque binaire représentant les éléments importants. Nous étudions également les performances des modèles avec un nombre différent d'exemples annotés et avec des différentes profondeurs de modèle. Ce travail a été présenté dans un article [38] dans le cadre de *International Conference on Computer Vision Theory and Applications, 2020.*

L'étape de formation consiste à extraire des caractéristiques des images avec barre de sécurité fermée, qui sont similaires aux caractéristiques du masque avec barre de sécurité fermée, mais différentes des caractéristiques du masque avec barre de sécurité ouverte (et l'inverse pour les caractéristiques extraites des images avec barre de sécurité ouverte). Lors de l'étape de test, nous extrayons simplement les caractéristiques de chaque image et vérifions si elles sont plus proches des caractéristiques du masque avec barre de sécurité ouverte ou du masque avec barre de sécurité fermée. Les résultats expérimentaux montrent que cette architecture est capable d'extraire des caractéristiques spécifiques de chaque télésiège. En effet, un seul réseau siamois formé sur 21 télésièges différents donne de très bons résultats sur chacun de ces télésièges.

B.2.6 Chapitre 6

Dans ce chapitre, nous proposons une solution pour améliorer le processus d'apprentissage d'un réseau de classification lorsque des images moins étiquetées sont nécessaires. Nous montrons que ce processus d'apprentissage original d'une architecture siamoise donne au modèle un pouvoir de généralisation important. Cette approche permet de remettre en question notre modèle en le formant et en le testant sur différents domaines (source et cible) et en montrant des résultats prometteurs. De plus, nous proposons une étape de raffinement en utilisant seulement deux masques binaires du domaine cible.

Une fonction de perte contrastive est utilisée pour fournir des propriétés euclidiennes à l'espace. Afin d'obtenir de bonnes propriétés de généralisation du modèle de classification des images d'un domaine inconnu, nous utilisons deux masques virtuels calculés dans l'espace de représentation (embedding), comme le centroïde des masques sources. Nous montrons que cette approche est plus performante dans le domaine cible qu'un classificateur d'images générique. Enfin, si des masques cibles sont disponibles, nous avons proposé une étape simple de raffinement qui contribue à améliorer la propriété de généralisation du modèle, en concentrant les images de différents télésièges autour des mêmes masques binaires. Au moment du test, la classification peut être réalisée en utilisant une simple distance euclidienne dans l'espace de représentation.

B.3 Conclusion et perspectives

Cette thèse présente notre travail sur le problème de la sécurité des télésièges en utilisant des techniques d'apprentissage profond. Dans le cadre du projet Mivao, nous visons à assurer la sécurité des télésièges en surveillant la station d'embarquement et en détectant les situations à risque pour prendre des mesures correctives avant qu'un accident ne se produise. À partir d'une image montrant des personnes montant dans un télésiège, nous avons proposé un ensemble de solutions pour classer la situation comme sûre ou non sûre. Comme il n'existe pas d'ensemble de données publiques contenant des images aussi spécifiques, nous avons dû proposer des modèles qui peuvent être formés avec peu de données annotées. En outre, deux problèmes principaux ont dû être abordés dans ce projet. Premièrement, puisque la propriété de sécurité est fortement liée à la position de la barre de sécurité, les modèles proposés doivent se concentrer sur des zones petites mais cruciales dans les images. Deuxièmement, en raison de la difficulté d'étiqueter les nouvelles images lorsqu'un système est installé sur un nouveau télésiège jamais vu, les modèles formés sur un ensemble de télésièges devraient donner de bons résultats sur un nouveau télésiège sans avoir besoin d'images étiquetées.

Notre première contribution est de montrer qu'une approche de détection d'objets peut être plus performante qu'une approche de classification d'images. Nous avons constaté que la détection d'objets aide le réseau à se concentrer sur les détails cruciaux dans les images. Les informations sur la localisation des objets peuvent être utiles pour pousser plus loin la classification d'images, en particulier dans la généralisation de ces approches.

Notre deuxième contribution consiste à proposer une composante d'adaptation du domaine au niveau de la proposition de région d'un détecteur à deux étages. Nous avons montré que l'adaptation est nécessaire au niveau de la proposition de région pour fournir de meilleurs résultats de détection dans le domaine cible où aucune étiquette n'est disponible, et cette solution originale a surpassé les solutions classiques.

Notre troisième contribution est de proposer un modèle siamois pour insérer quelques antécédents géométriques dans un réseau de classification d'images. Nous avons montré qu'en utilisant une architecture siamoise, le réseau pouvait s'occuper des éléments les plus cruciaux de la scène. Notre réseau siamois a pris comme entrée une paire d'images: une image colorée de la scène et une image binaire représentant le masque de la barre de sécurité. Seuls deux masques étaient nécessaires pour chaque télésiège. Cette approche a prouvé son efficacité même avec peu de données étiquetées et a surpassé un classificateur d'images classique.

Notre contribution finale est un ensemble de méthodes pour évaluer et renforcer la propriété de généralisation du modèle siamois. Nous avons montré que notre réseau siamois avait des propriétés de généralisation importantes, indiquant qu'à mesure que les domaines d'intérêt s'étendaient, l'accent sur les régions visuelles les plus cruciales de la scène devenait plus important. Un modèle unique formé sur un ensemble de télésièges a donné des résultats exceptionnels sur un nouveau télésiège jamais vu, sans avoir la nécessité d'aucune étape d'adaptation. Il est à noter que ce dernier modèle a surpassé le détecteur d'objets adaptatif.

Il existe de nombreuses directions potentielles qui doivent encore être explorées pour améliorer nos différentes contributions. Quelques travaux futurs possibles sont présentés ci-dessous. Les résultats de Faster R-CNN pourraient être améliorés en utilisant des architectures profondes avancées pour extraire des caractéristiques comme ResNet. Une perspective est d'évaluer la détection des personnes et des barres de sécurité dans un système unifié.

En outre, des procédures d'adaptation plus rigoureuses pourraient également améliorer les résultats de l'adaptation du domaine pour la détection d'objets. Ces méthodes pourraient contribuer à l'étape d'apprentissage pour parvenir à des solutions stables, ce qui constitue une faiblesse importante des méthodes actuelles. Enfin, nous pensons qu'un nouveau processus de renforcement du modèle siamois utilisant uniquement les masques virtuels, améliorerait encore la généralisation. Une autre perspective pour valider la performance de cette approche serait de la tester dans une autre application différente de la sécurité des télésièges.

Nous nous sommes attaqués au problème de la sécurité des télésièges, tant en ce qui concerne la détection des objets que la classification d'images. La question suivante se pose tout naturellement: serait-il utile d'analyser des données vidéo plutôt que des données d'images? Serait-il avantageux d'utiliser des informations temporelles et d'aborder le problème comme un suivi d'objet plutôt que comme une détection d'objet?