UNIVERSIDADE FEDERAL FLUMINENSE

LUIGY ALEX MACHACA ARCANA

TrADe Re-ID – Improving Person Re-Identification using Tracking and Anomaly Detection

NITERÓI

2022

LUIGY ALEX MACHACA ARCANA

TrADe Re-ID – Improving Person Re-Identification using Tracking and Anomaly Detection

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

Orientador:
Prof. Dr. Esteban Walter Gonzalez Clua

Coorientador:
Prof Dr. Joris Michel Gérard Daniel Guerin

NITERÓI

2022

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

A668t Arcana, Luigy Alex Machaca
TrADe Re-ID - Improving Person Re-Identification using Tracking and Anomaly Detection / Luigy Alex Machaca Arcana. - 2022.
62 f.: il.

Orientador: Esteban Walter Gonzalez Clua.
Coorientador: Joris Michel Gérard Daniel Guerin.
Dissertação (mestrado)-Universidade Federal Fluminense, Instituto de Computação, Niterói, 2022.

1. Live Person Re-Identification. 2. Tracking. 3. Anomaly Detection. 4. Produção intelectual. I. Clua, Esteban Walter Gonzalez, orientador. II. Guerin, Joris Michel Gérard Daniel, coorientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

LUIGY ALEX MACHACA ARCANA

TrADe Re-ID – Improving Person Re-Identification using Tracking and Anomaly Detection

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

Aprovada em Novembro de 2022.

BANCA EXAMINADORA

Prof. Dr. Esteban Walter Gonzalez Clua - Orientador, UFF

Prof. Dr. Joris Gérard Daniel Guerin, Co-orientador, UFF

Prof. Dra. Aura Conci, UFF

Prof. Dr. Luiz Marcos Gonçalves, UFRN

Niterói

2022

Á minha família toda, em especial para minha Mãe Rosa e meu pai no céu Rolando pelo apoio e dedicação.

Agradecimentos

A professor Esteban Clua e Joris Guerin meus orientadores, por sua paciência e motivaçao neste caminho da pesquisa, todo meu carinho e reconhecimiento por me auxiliar no desenvolvimento deste trabalho.

Aos profesores do Instituto da Computação (IC) da UFF por compartilhar seu conhecimento que permitiram aprofundar ao longo do meu periodo de estudos.

Aos meus pais, minha Mãe Rosa e meu pai no céu Rolando, pela sua incansável dedicaçao e apoio em todas as etapas de minha vida. Á meus irmãos por acreditar em mim.

A minha querida Maria, pela imensa paciência nos dias mais difíceis, pelo apoio incondicional e suporte emocional nesta etapa de minha vida.

Por fim, aos meus amigos, pelos laços de amizade criados e que tornaram uma estada agradavel no Brazil.

Resumo

A Re-Identificação (Re-ID) de Pessoas é um problema de Visão Computacional, cujo objetivo é procurar uma pessoa de interesse (query) em uma rede de câmeras. No cenário do Classic Re-ID a consulta é procurada em uma galeria pronta contendo imagens devidamente segmentadas de corpos humanos inteiros. Recentemente foi introduzida uma nova configuração do Live Re-ID para representar melhor o contexto prático de aplicação da Re-ID. Esta abordagem consiste na busca da consulta em vídeos curtos, contendo quadros inteiros da cena. A base inicial que foi proposta para abordar o Live Re-ID usou um detector de pedestres para construir uma grande galeria de busca a partir do vídeo, e aplicou um modelo clássicc Re-ID para encontrar a query na galeria. Entretanto, as galerias geradas eram muito grandes e continham imagens de baixa qualidade, o que diminui significativamente o desempenho do Live Re-ID. Neste trabalho apresentamos uma nova abordagem do Live Re-ID chamada TrADe, para gerar galerias pequenas e de alta qualidade. TrADe primeiro usa um algoritmo de Tracking para gerar tracklets (seqüência de imagens do mesmo indivíduo) na galeria. Em seguida, um modelo de Anomaly Detection é usado para selecionar um único melhor representante de cada tracklet. Validamos a eficiência do TrADe e Live Re-ID na dataset PRID-2011, e mostramos melhorias significativas em relação à base inicial.

Palavras-chave: Live Person Re-Identification, Tracking, Anomaly Detection.

Abstract

Person Re-Identification (Re-ID) is a computer vision problem, which goal is to search for a person of interest (query) in a network of cameras. In the classic Re-ID setting, the query is sought in a curated gallery containing properly cropped images of entire human bodies. Recently, the live Re-ID configuration was introduced to represent better the practical application context of Re-ID. It consists in searching for the query in short videos, containing whole scene frames. The initial baseline proposed to address live Re-ID used a pedestrian detector to build a large search gallery from the video, and applied a classic Re-ID model to find the query in the gallery. However, the galleries generated were too large and contained low-quality images, which decreased the live Re-ID performance significantly. In this work, we present a new live Re-ID approach called TrADe, to generate smaller high-quality galleries. TrADe first uses a Tracking algorithm to identify tracklets (sequences of images of the same individual) in the gallery. Following, an Anomaly Detection model is used to select a single representative of each tracklet. We validate the efficiency of TrADe on the live Re-ID version of the PRID-2011 dataset and show significant improvements over the initial baseline.

Keywords: Live Person Re-Identification, Tracking, Anomaly Detection.

List of Figures

List of Tables

Contents

1 Introduction

1.1 Context and Motivation

Video surveillance systems are now widely used in public places [1]. These systems consist of a network of cameras strategically positioned to be monitored by human security agents for public safety [2, 3].

The growing urbanization and the improvement of information channels (i.e., Telecommunications networks) are the most important phenomena influencing the shift in planning paradigms toward urban safety, the sustainable use of natural resources, and adequate public spaces [1, 4]. Hence, the demand for real-time automated pedestrian tracking systems is rapidly increasing.

Although human agents are able to analyze precisely any given scene, they lack the possibility to monitor a large number of cameras simultaneously [5, 6]. Nonetheless, Computer vision seeks to reproduce the human's ability of analysis to build an automated pedestrian tracking system to facilitate activities of monitoring.

This work deals with the Person Re-Identification (Re-ID) problem, consisting in searching for a person of interest (query) in a network of non-overlapping cameras. The goal of Re-ID is to tell whether the query was observed in one of the cameras during a given period [7].

The most common setting for Re-ID uses datasets of cropped images of humans, collected from such a network of cameras and manually curated to ensure that it contains only clean full-body images. Then, the objective is to retrieve the images from the search gallery that correspond to the same individual as the query image [8]. In this work, we refer to this setting as classic Re-ID.

Recent works have shown that the classic Re-ID setting is not sufficient to implement useful real-world applications of person re-identification. In our previous work, we have examined the effects of object detectors and person re-identification to propose a novel

setting called Live Re-ID [9] sets a clear and reliable baseline on Live Re-ID setting. This technology has various potential applications, such as suspect searching [10], identifying owners of abandoned luggage [11], and recovering missing children [12]. This new setting considers constraints related to the implementation of Re-ID applications for use during live operations. Live Re-ID systems are composed of two main modules: the gallery generator, which extracts pedestrian bounding boxes, and the classic Re-ID module, which tries to identify the query from the cropped images in the gallery (see Section 2 for more detail).

1.2 Problem Statement

Although most Re-ID research has focused on the classic Re-ID module, our experiments [9] demonstrated that small errors in the gallery generation process can lead to poor Live Re-ID results. This is illustrated in Figure 1, which represents all the cropped bounding boxes generated with the baseline Live Re-ID system proposed in our previous work [9]. In the baseline Live Re-ID setting, only a pedestrian detection model is used to generate a massive search gallery (4,271 cropped images) containing every detected bounding box.

Furthermore, the main limitation identified in this previous work for successful Live Re-ID implementations are:

1. the fact that the object detector used for gallery generation sometimes generates bad bounding boxes; in other words, these bounding boxes correspond to incomplete body parts (i.e., legs, arms, or torso) which not represent entire human bodies,
2. the fact that it generates massive galleries, containing many correlated bounding boxes representing the same individuals, and
3. the fact that massive galleries, impact both the accuracy and execution time of the subsequent Re-ID module.

On the other hand, approaches from the field of video-based Re-ID have shown that using sequences of consecutive images of the same person can be valuable for Re-ID performance [14]. Indeed, videos include much richer data than single images as we know that bounding boxes close to each other in space and time are likely to represent the same person. For example, in Figure 1 we can see that the standard gallery generation module generates a large number of bounding boxes, including poorly cropped ones near the

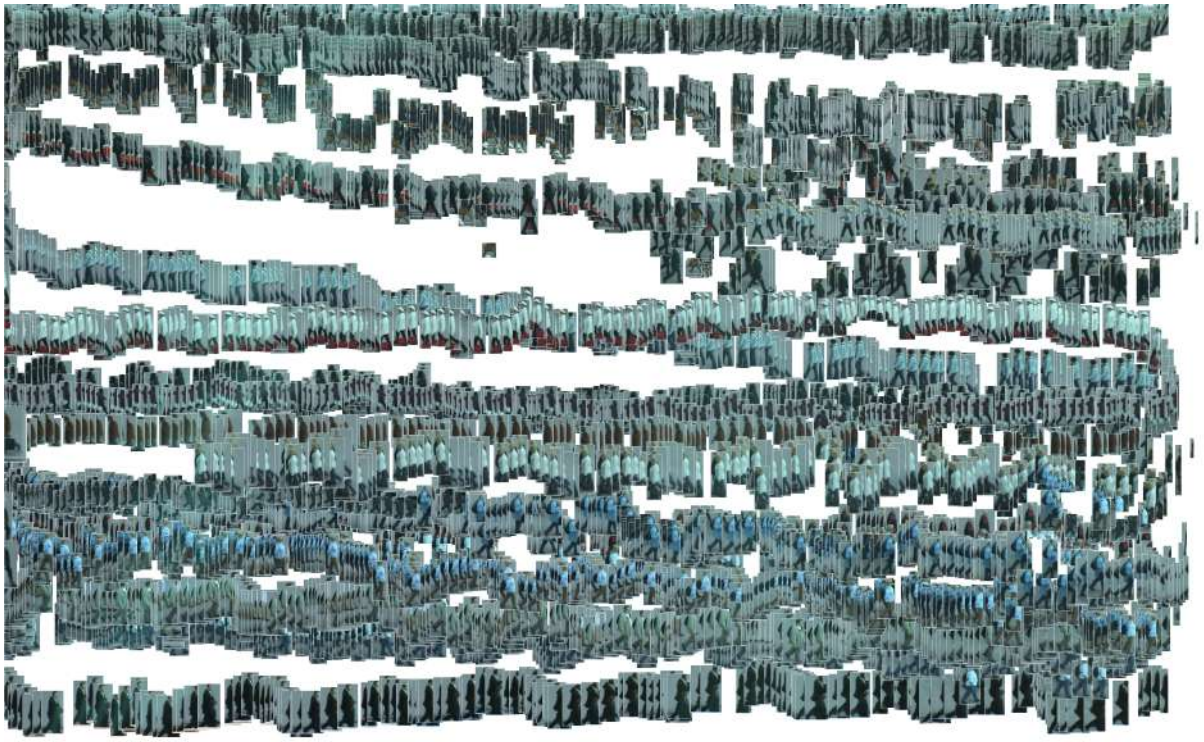


Figure 1: Gallery generated by standard Live Re-ID. We applied this approach to a ~ 2 minute video from the PRID [13] dataset

edges. However, using Tracking, we can gain information and recover tracks representing the same individuals (Figure 6).

In order to solve the above problem, we propose a novel Live Re-ID approach to considerably reduce the size of the gallery that is produced by the gallery generation module. This approach is called TrADe (gallery filtering using **Tr**acking and **A**nomaly **De**tection).

1.3 TrADe Overview

TrADe, a new approach of Live Re-ID, uses an object tracking algorithm [15] to identify tracklets (consecutive bounding boxes corresponding to the same individual as we shown in Figure 6(a & b)), and an anomaly detection algorithm to select a single good representative image of each tracklet (Figure 6(c)).

Figure 13 shows the steps involved in our proposed pipeline for Live Re-ID, which uses the same version as our previous work, You Only Look Once version 3, YOLOv3 [16] as an object detection model; a Real-Time Recurrent Regression, Re3 [15] as a visual object tracking model; a Deep One-Class classification, DOC [17] as a one-class classifier

for anomaly detection, and tests two different approaches for classic Re-ID.

By decreasing drastically the gallery size and removing bad images, TrADe allows us to improve the accuracy of the whole Live Re-ID system, and reduce the execution time of the Classic Re-ID module. We conduct experiments on the same Live Re-ID Dataset as Sumari et al. [9] and show that TrADe outperforms both their baseline approach and another simple baseline approach for gallery filtering.

1.4 Research Objectives

Sumari et al. [9] set a clear and reliable baseline on Live Re-ID (Section 1.1). In the present research, we go a step further in the field of Live Re-ID. As we mentioned before, Live Re-ID is composed of two main modules: the gallery generator and the classic Re-ID module. We focus to improve the module of gallery generation of Live Re-ID and optimization of the classic Re-ID module.

1.4.1 Main Objective

The main objective of this work is to decrease the gallery dataset size drastically and remove bad images from the gallery generator of Live Re-ID. By doing this, we hope to achieve two specific purposes that together achieve the overall goal of this dissertation, as follows:

1. Improve the accuracy of the whole Live Re-ID system, and
2. Reduce the execution time of the classic Re-ID module.

1.4.2 Contributions

Our results were accepted in ICMLA (International Conference on Machine Learning and Applications) [18], where our main contribution is the proposal of a novel pipeline for the Live Re-ID problem, called TrADe, decreasing drastically the gallery size and removing bad images, and improve the accuracy of the whole Live Re-ID system in order to implement and evaluate real-world security applications.

Furthermore, experiments are conducted to demonstrate the importance of TrADe to tackle real situations. The evaluation of the TrADe pipeline is conducted with the same dataset and evaluation metrics as Sumari et al. [9].

This research does not claim to have solved the Live Re-ID, it is getting closer of a solution in practical scenarios. We hope to inspire and motivate the community to focus on the Live Re-ID systems, in order to develop algorithms that are better adapted for real-world scenarios.

1.5 Dissertation Organization

The rest of this document is organized as follows: In Chapter 2, we present relevant related work. Chapter 3 describes some important definitions to understand in detail our proposal and comparison to the baseline, such as object detection, object tracking, One-Class Classification, and Person Re-identification. In Chapter 4 we present a new Live Re-ID approach called TrADe and justify its applicability to real situations. Also, this chapter describes the architecture and components of the pipeline system. In Chapter 5 we present results and discussion about our solution. Finally, Chapter 6 presents conclusions and possible future work for this dissertation.

2 Related work

In this chapter we present and review the most relevant works that supports and relates to our research, in order to generate a good comprehension of our novel Live Re-ID approach. Also, we present the state-of-the-art about different topics that are used in this work. The algorithms used in this document are based on Deep Learning.

In recent years, the rapid development of techniques based on deep learning [19], played an essential role in achieving good results at various Computer Vision tasks that deal with detection, tracking, and re-identification instances of visual objects of a particular class (e.g., pedestrians, cars, etc.). Thus, it has been widely used in many areas, such as object detection, object tracking, anomaly detection, and person re-identification, where some of them even achieved real-time applications.

2.1 Deep Learning in Computer vision

Computer Vision (CV) is a field of Computer Science allowing computer systems to consistently extract information from digital visual inputs (i.e. images, videos). Computer vision techniques provide computers with the ability to observe and understand in the same way as humans do, so that they can achieve meaningful and coherent actions such as detecting, localizing and classifying different objects.

Deep Learning (DL) is a subfield of machine learning methods that allows computational models learn representations of data with multiple levels of abstraction [19] to build artificial intelligence systems based on a family of functions inspired by the human brain, called Artificial Neural Networks (ANN). A DL architecture is commonly formed by multiple layers of non-linear processing stages [20], where the input of the lower layer is fed with the output of immediate high layer and the first layer is the input data (e.g. image, video). As a result, the lower-level features (from low layers) gradually merge to form higher-layer features. A most representative of classic DL architecture is called Convolutional Neural Network (CNN) which uses intermediate layers, such as convolutional

layers, pooling layers, Rectified Linear Unit (ReLU) layers, and fully connected layers [21].

In recent years, Deep Learning has grown up quickly [22] and exhibited a strong advantage in high-level abstractions over data. The current applications of Deep Learning cover areas such as computer vision, natural language processing, and sound analysis. Chai et al. [23] provides a review of advances in Deep Learning, and shows how Deep Learning covered these domains.

Computer Vision and Deep Learning perform jointly to emulate human vision to create computer vision systems in order to facilitate observing an area through surveillance cameras, or security cameras. These computer vision systems aim to tell apart and comprehend visual inputs to operate according to a particular situation automatically.

2.2 Object Detection

Object detection is a computer vision problem, which objective is to locate the instances of an object of interest in digital images. It consists in answer the following question: "what visual objects are there, and where are they?". Object detection was already discussed extensively in recent surveys [24, 25], presenting the most advanced and fastest methods of detecting different types of objects to simulate human vision and cognition. These instances can be divided into disjoint categories (e.g., person, car), and each of them is represented with a bounding box.

In Figure 2, we show object detection results obtained from applying it to Oxford-Town [26] Centre Dataset. Where the color boxes represent only the person class predictions. For easier interpretation, we assign a number over each box and not a traditional score. A bounding box (BB) detection is considered correct if the output bounding box has a sufficiently large overlap with the ground truth bounding box. This is evaluated with a metric called Intersection over Union (IoU).

In Figure 3, we illustrate how to apply IoU to evaluate any object detector algorithm, for this purpose, we need: (1) The ground-truth bounding boxes (provided with labeled dataset such as PASCAL VOC), (2) The predicted bounding boxes (outcome of the algorithm. Additionally, in Figure 4, we include visual examples for some scores of IoU. IoU's score is a number from $0 to 100\%$ that determines the portion of overlap between ground truth and the predicted bounding box. Where 0 means that there is no overlap (**Failure**), and values near 1 means both bounding boxes are almost totally overlapping, i.e., ground truth and predicted bounding boxes are almost the same (**Very Good**). For

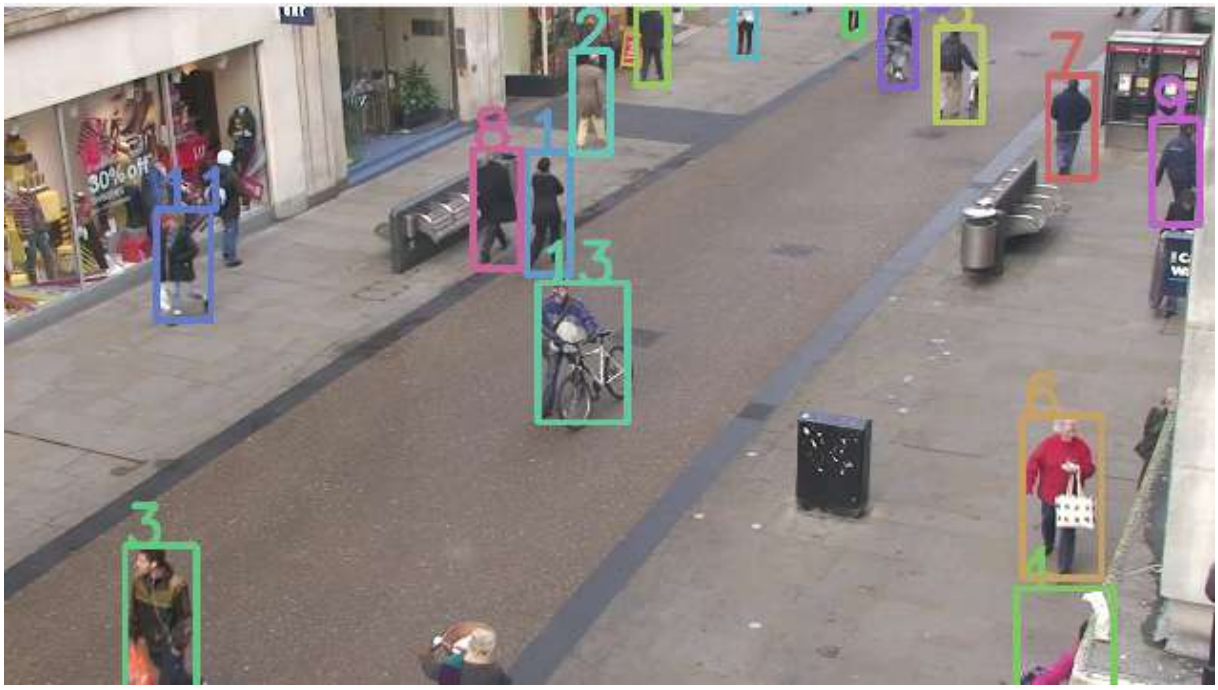


Figure 2: Object detection results where color boxes represent only the person class predictions.

our purposes, we use only IoU's values greater than or equal to 50% (**Good**).

As we mentioned above, IoU is an evaluation metric that is used in object detection challenges such as PASCAL VOC [27]. Thus, IoU metric is used to measure the accuracy of an arbitrary object detector algorithm, i.e., any algorithm that provides predicted BBs as an outcome can be estimated employing IoU. Typically, we consider that a predicted BB is correct if it has an IoU greater than 50%.

The current object detection methods can be divided into two predominant lines: one-stage approaches (e.g. SSD [28], YOLO [29]) and two-stage approaches (e.g. Faster R-CNN [30]). The one-stage detectors perform higher inference speed, while the two-stage detectors have higher localization and object recognition accuracy.

In order to explain the two-stage approaches, we use the Faster R-CNN [30] architecture as an example. The first stage proposes candidate object bounding boxes (BBs) called Region Proposal Network RPN. The second stage processes each candidate by Region of Interest RoI pooling layers. Each feature is processed for the following classification and BB regression tasks [30].

Regarding the one-stage detectors, also called Unified Detectors, they refer to architecture that directly predicts class probabilities and bounding boxes from full images with a single feed-forward over Convolutional Neural Network (CNN). For this reason, they

are the fastest methods for detecting objects and assigning class categories. In particular, You Only Look Once (YOLO) was proposed in 2015 by Redmon et al. [29]. Encapsulating all computation in a single network. Additionally, Redmon has made improvements based on YOLO and proposed its v2 [31] and v3 [16] editions.

In this work, we use object detection specifically for detecting persons. This problem is called pedestrian detection.

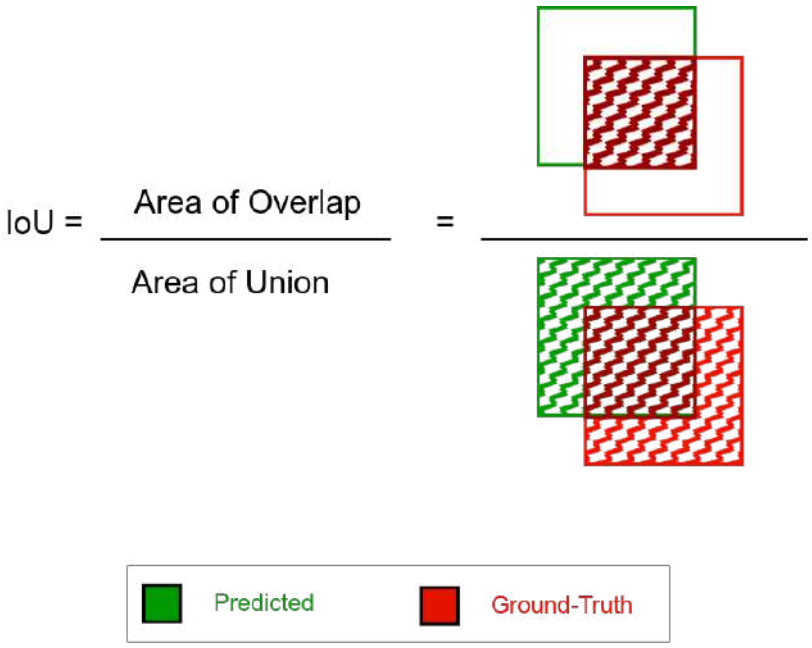


Figure 3: Intersection over Union (IoU) formula.

2.3 Object Tracking

Object tracking is another problem within the field of computer vision. It seeks to enable a computer to replicate the basic functions of human vision [32], such as motion perception and scene understanding. Object tracking can be applied to many domains [33] such as surveillance, human-computer interaction, etc.

The main task of object tracking is to establish the location of a target object over the frames of a video sequence, starting from an initial bounding box [34].

In the literature, several surveys have proposed different classifications of object tracking approaches, e.g., single-camera vs. multi-camera Tracking [35], single-object vs. multi-object tracking [36], specific vs. generic object tracking [24]. [37, 35] and state-of-art challenges (e.g., Visual Object Tracking, Multi-Object Tracking) [38, 36, 3].

Furthermore, the majority of current object trackers are trained over known object

types or specific object instances [39], i.e., pedestrians, cars, etc. However, sometimes it is not feasible to pre-specify what kind of object needs to be tracked, and we simply want a remote user to specify an object of interest by clicking on a single image frame. To address these problems, a new task called Generic Object Tracking [24]) was developed.

In this work, we use an algorithm called Real-time, Recurrent, Regression-based tracker (Re3) [15], which is an accurate network for Generic Object Tracking. Re3 uses convolutional layers to embed the object appearance, recurrent layers to recall the appearance and motion of the object information, and a regression layer to output the object location. Re3 requires a bounding box around tracked objects at initial time-step T_0 and produces bounding boxes for the object in subsequent frames.

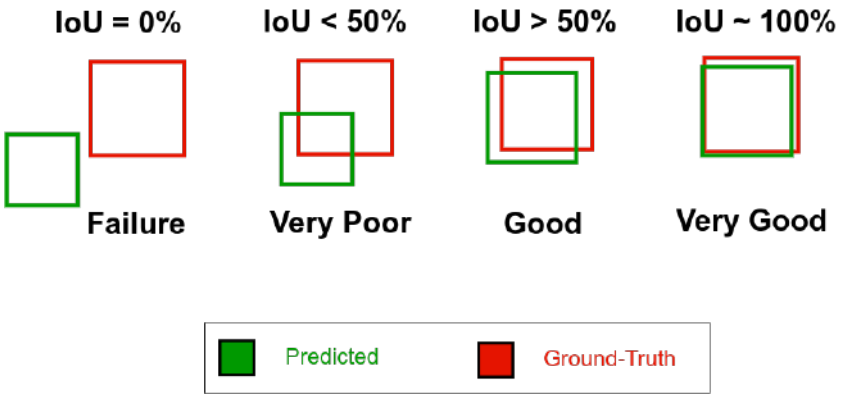


Figure 4: Intersection over Union Scores.

2.4 Anomaly Detection

Anomaly detection (also called outlier or novelty detection) refers to a wide research problem of identifying data that significantly diverge from the patterns of expected data instances [40, 41]. Anomaly detection can be applied in domains such as security, surveillance, and AI safety [42, 43].

In recent years, deep learning approaches were used extensively to tackle anomaly detection. Deep anomaly detection models can be split into supervised, semi-supervised, hybrid, unsupervised, and end-to-end, as shown in recent surveys [42, 40, 43].

Additionally, anomaly detectors rely on the assumption that the model extracts good abstraction features from data input, which keeps discriminative information that separates anomalies from regular instances.

Anomaly detectors are composed of two networks, a feature extractor to retrieve

discriminative information that separates anomalies from regular instances, and a classifier [17, 44]. And, the objective of the classifier is to detect all the non-native-class data, resulting in incorrect class predictions [43]. In the context of this work, we called a non-native class all classes that were not used for training. Thus, the non-native class may be called a novel or abnormal class detection, depending on the application domain.

A subfield of Anomaly Detection technique is called One-Class Classification (OCC), which consists in defining a classification boundary around the native class (normal instances). At inference stage, an OCC model generates a "normality" score that can be used to determine if it is an inlier or an outlier. In this work, we use an efficient recent approach for OCC, called DOC (Learning Deep Features for One-Class Classification) [17].

DOC takes advantage of a CNN for feature extraction, using two loss functions for training:

1. The compactness loss fosters low intra-class distances by evaluating the closeness of the native class among the learned space features.

2. The descriptiveness loss aims at finding large inter-class distances.

This feature extraction network is trained on a dataset containing images from both native and non-native classes. Then, DOC uses a second neural network to produce the final classification score.

2.5 Person Re-identification

Person Re-Identification (Re-ID) is a fundamental problem to be solved in Re-ID systems for automated video surveillance due to a critical request for public safety and the increasing number of surveillance network cameras [14, 45] at places such as streets, squares, parks, among others.

Re-ID aims to re-identify a person of interest (PoI) across multiple non-overlapping cameras [10], i.e., given a PoI that can be an image (or video) from one camera, Re-ID has to search and mark the PoI across other cameras (or even the same camera) at a different time, instant, and place.

In other words, Re-ID consists in retrieving instances of an individual, called the query, within a set of complex multimedia content, called the gallery. The most popular Re-ID setting, now denominated as classic Re-ID, consists of representing both the query

and all the items in the gallery with well-cropped images representing a person's entire body (see survey [46]).

Re-ID methods can be categorized into two main threads according to [45, 10] closed-world and open-world Re-ID settings:

1. The closed Re-ID setting is usually applied under the following assumptions [14, 45]: (1) Person appearances are extracted from a single modality visible camera. (2) Training and testing based on bounding boxes that mainly contain a person's appearance information. (3) The bounding box annotations are generally correct and must appear in the gallery set.

2. On the contrary, the open Re-ID setting works with heterogeneous data from multiple cameras, bounding boxes taken directly from raw images/videos, and bounding boxes usually bring a noisy annotation.

The person search setting uses galleries composed of whole scene images, to represent the real-world application context of Re-ID better. Hence, a person search approach must return both the index of the image where the query is present and its location in the image (see survey [47]).

The open-set Re-ID setting extends classic Re-ID by adding the option that the query is not present in the gallery (see survey [48]). Besides, in the video-based Re-ID setting, the query and gallery images are replaced by sequences of consecutive images from a video. Sequences contain clean full-body images representing the same individual (see survey [8]).

2.6 Live Re-ID

More recently, the Live Re-ID setting was introduced by Sumari et al. [9] to represent relevant aspects for deploying Re-ID in real-world applications:

1. During live operations, the entire raw stream of the video frames must be used as input.

2. The predictions by the Re-ID system are then verified by a human monitoring agent.

To tackle these it was necessary to formalize a new setting between the Re-ID system and the monitoring agent to ensure natural interaction. This setting combines elements

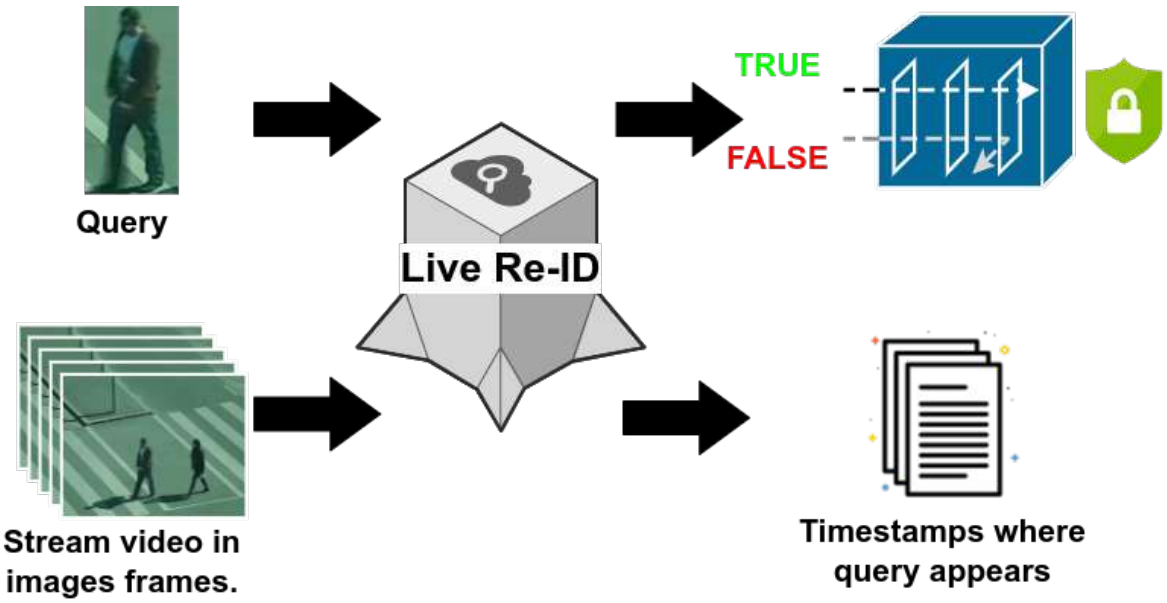


Figure 5: Live Re-ID flow.

from several of the Re-ID settings mentioned above and was further formalized in [49]. In practice, finding a query person during live operations requires processing whole scene videos in near real-time. In Figure 5, we depict how Live Re-ID determines if the query is present in the streaming video. And, when and where it appeared.

This way, the galleries for Live Re-ID contain whole scene video frames. In addition, the probability that the query is present in a short video sequence from a given camera is low, which means that the Live Re-ID setting is open-set. Finally, Live Re-ID also accounts for the fact that Re-ID model predictions must be verified by human security agents, who can trigger actions. Hence, new evaluation metrics were proposed to evaluate two objectives:

1. High re-identification rate, i.e., enhance the results when it was successfully identified a person sought by Live Re-ID system, and

2. Low false alarm rate, i.e., reduce the case where operator agent is disturbed for nothing.

In this work, we propose a new approach to generate better image galleries within a Live Re-ID context. Our objective is to show that generating smaller galleries of higher-quality images can substantially improve Live Re-ID results, even without changing the classic Re-ID models used for final re-identification. In our experiments, two classic Re-ID models are tested:

1. The Bag of Tricks (BoT) approach is based on several neural network training tricks (e.g., controlling hyperparameters, using both triplet and cross-entropy loss, using the Adam optimizer) rather than Re-ID architectural choices [50].

2. The Siamese Improve Deep Learning (SiamIDL) approach uses a Siamese neural network architecture to evaluate image similarity and predict whether two input images represent the same person [51].

3 Background

In this chapter, we discuss the specific concepts used in this work. Section 3.1 introduces the formal concept of tracklet. Section 3.2 introduces the concept of You Only Look Once (YOLO). Section 3.3 introduces the main concepts of Real-Time Recurrent Regression Network (Re3). Section 3.4 shows a Deep One-Class (DOC) classification method. Finally, in sections 3.5 and 3.6 we introduce the main concepts of a person re-identification used in this document.

3.1 Tracklet

The concept of a tracklet was discussed in several areas of interest for this work, such as object detection, tracking detection, and person re-identification. Intuitively a tracklet can be defined as a short track or part of a complete track. We illustrate both a complete track in Figure 6(a) and tracklets in Figure 6(b).

This definition of tracklets has been used in different works, such as:

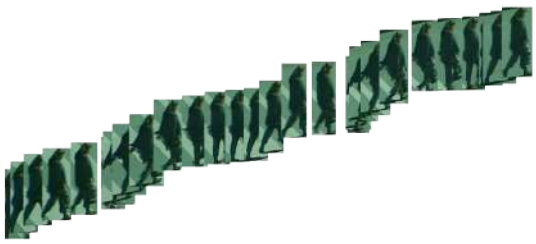
1. The necessity to gather detections corresponding to the same individuals from a set of detections from consecutive frames to produce tracklets was discussed by Brendel, Amer, and Todorovic [52].

2. Brendel, Amer, and Todorovic [52] used tracklets based on a graph to take advantage of graph properties to find complete trajectories.

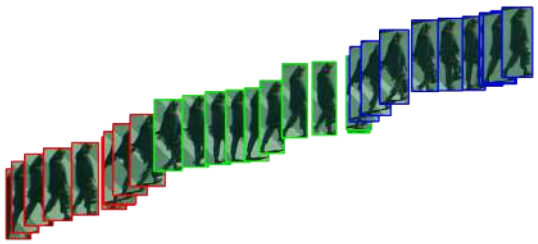
3. Wang et al. [53] used tracklets to produce effective and complete trajectories even if they are spatially close or occluded.

4. Zhang et al. [54] used clustering over set of tracklets to generate entire trajectories.

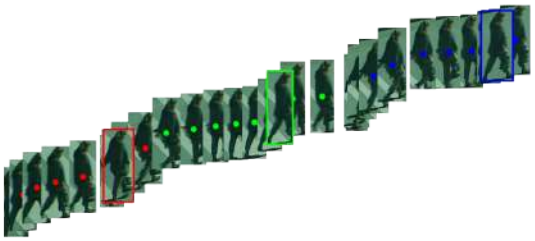
5. Cheng et al. [55] discussed how to find associations to complete full trajectories from tracklets. They generated a gallery of tracklets from videos from non-overlapping uncalibrated cameras.



(a) Example track generated by the Re3 tracking algorithm.



(b) The track is divided into small fixed-size tracklets.



(c) One candidate is selected for each tracklet

Figure 6: Track and Tracklets. (a)The entire trajectory of a person is found using the Re3 tracking algorithm. Then, (b) the full track is divided into small fixed-size tracklets. (c)Finally, we apply DOC over each tracklet to select a good representative image.

Also, it is common for people to walk through areas where there are cameras present, such as in public spaces, retail stores, and office buildings. As a consequence, thousands of unlabeled Re-ID public videos are generated everyday. Li, Zhu, and Gong [56] proposed that each video can be used to generate new Re-identification datasets using tracklets to conduct more comprehensive evaluations and analyses.

In this work, the concept of tracklet works under the following assumptions:

1. It is very likely that each tracklet in the same video represents a different person.
2. Each tracklet is unique even if the same person is in different cameras due to such things as adverse weather and illumination conditions (e.g, rain, low-light, night-time, and shadows or pathways).
3. Each bounding box of tracklet contains mostly the same person.

We understand a tracklet as a list of object bounding-boxes, as we illustrated in Figure 6(b), which has three tracklets that belong to the complete trajectory of a person, this example illustrates how tracklets can come together to form a complete track.

Furthermore, a tracklet can be defined as a short-term trajectory of one target object across frame sequences on batch video:

$$\mathbf{T}_{(k,N)} = \left\{\mathbf{b}_t^k, \mathbf{b}_{t+1}^k, \mathbf{b}_{t+2}^k, \cdots, \mathbf{b}_{t+(N-1)}^k\right\} \tag{3.1}$$

Where $\boldsymbol{k}$ represents the ID number label for all bounding-boxes that compose a tracklet, $\boldsymbol{t}$ is a ID number of frame within the video, $\boldsymbol{N}$ is the maximum allowed field length for the fulfillment tracks (note: $\mathbf{T}$ can contain fewer elements than $\boldsymbol{N}$).

Also, we characterize a simple bounding-box within in specific frame as follows:

$$\mathbf{b}_t = (\mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_1, \mathbf{y}_1), \tag{3.2}$$

where the first pair $(\mathbf{x}_0, \mathbf{y}_0)$ represents the top-left and the second pair $(\mathbf{x}_1, \mathbf{y}_1)$ the bottom-right coordinates in the frame.

Moreover, we can denote a set of bounding boxes that belong to the same frame ($\boldsymbol{t}$), as follows:

$$\mathbf{B}_t = \left\{\mathbf{b}_t^1, \mathbf{b}_t^2, \mathbf{b}_t^3, \cdots, \mathbf{b}_t^m\right\} = \mathcal{D}_t \tag{3.3}$$

For this example, we can take an initial frame, $t = 0$, that represents initial bounding boxes at the beginning of the video ($\mathcal{D}_0$).

3.2 You Only Look Once (YOLO v3)

Object detection is a computer vision problem, which was already discussed in Section 2.2, where the goal of all methods is to reduce the efforts of humans to locate objects in images or videos [57].

In this section, we present You Only Look Once (YOLO), which first version was proposed in 2015 by Redmon et al. [29]. YOLO encapsulates all computation in a single network. Additionally, all following versions apply incremental improvements based on it. YOLO aims to locate object instances that belong to predefined categories in digital images, such as a person, a cat or a dog. With the aim of providing answers to the following questions:

- What visual objects are there?
- Where are these objects in the image?

YOLO is a Unified Detector framework, also called one-stage detector, that refers to

architectures that directly predict bounding boxes and class probabilities from full images, with a single forward pass over a Convolutional Neural Network. YOLO is powerful to identify even small objects from image inputs. Also, these advantages make it one of the most advanced and fastest methods to detect different types of objects to date.

YOLO divides an image input into $S \times S$ grids where each grid cell is responsible for the detection of objects. And also, each grid cell predicts a set of C conditional class probabilities, B bounding box locations, and confidence scores. We express these variables as one tensor, as follows:

$$S * S * (B * 5 + C) \tag{3.4}$$

Where S is the number of grid cell divisions, B represents the number of bounding boxes predicted by each grid cell, and C is a confidence score that indicates the probability that the bounding box contains an object. It can be calculated by Equation 3.5.

$$C = \Pr(\text{ Object }) * IoU_{\text{pred}}^{\text{truth}} \tag{3.5}$$

Where $\Pr(\text{ Object }) \geq 0$ indicates the probability that objects exist and $IoU_{\text{pred}}^{\text{truth}}$ indicates the IoU score between predict and ground-truth bounding boxes. Accordingly, C is zero if the grid cell does not contain an object because $\Pr(\text{ Object }) = 0$, otherwise, C is equal to $IoU_{\text{pred}}^{\text{truth}}$. Additionally, YOLO predicts a set of C conditional class probability (each conditional class probably belongs to only one object class) as was described by Redmon et al. [29].

In Figure 7, we illustrate how processing images with YOLO works with a single forward pass. YOLO divides an image input into $S \times S$ grids where each grid cell is responsible for the detection of objects. In other words, each grid cell predicts B bounding box (see Equation 3.2) locations, confidence scores for each of them, and C class probabilities. Finally, They are all part of a tensor, as we can see in Equation 3.4.

An important reason why we choose YOLOv3 [16] for our experiments is its enhanced multi-scale prediction capability compared to previous YOLO versions. This way, this third version can detect small objects even better. Additionally, the chosen backbone for YOLOv3 is Darknet-53, which uses 53 convolutional layers with 3x3 kernels in the beginning and 1x1 in the end (see Figure 8).

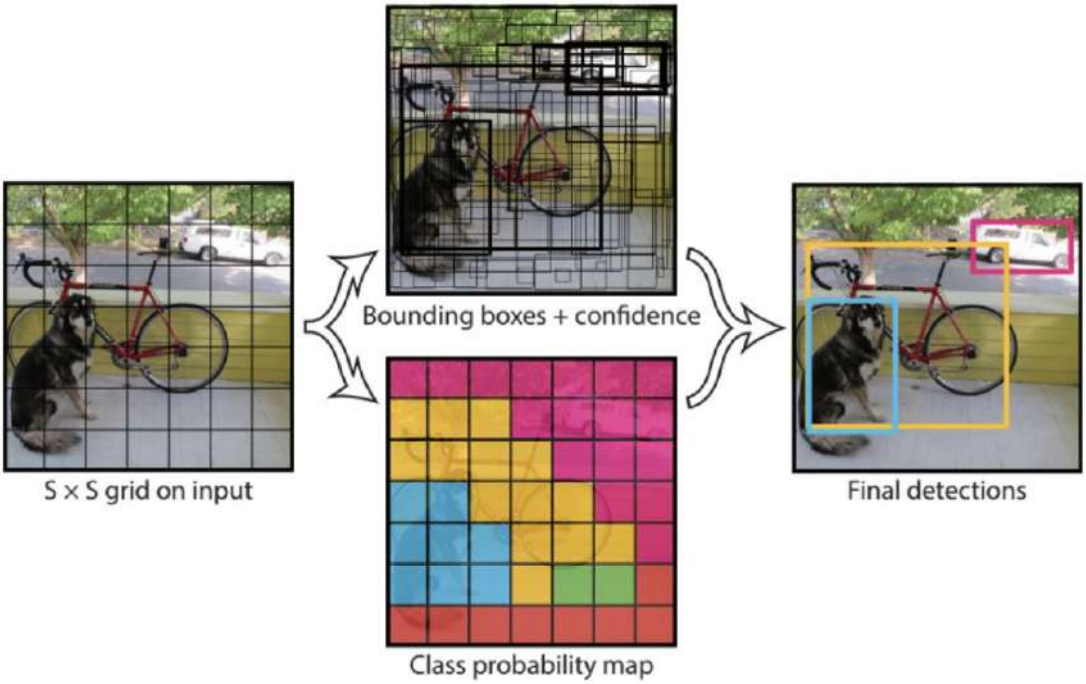


Figure 7: Yolo Flow, we adapted this image from [29].

3.2.1 Training Details

Multiple bounding boxes are predicted in each grid cell, as illustrated in Figure 7. Besides, at the training stage, each of the bounding boxes predicted is responsible for a unique object. To achieve this we use an IoU metric between a ground-truth and predicted bounding boxes, as illustrated in Figure 3. We only take the best IoU score.

The loss function used by YOLO is the sum of the Classification loss, the Localization Loss, and the Confidence Loss. We detail each of them:

1. **Classification Loss ($\mathcal{L}_{classif}$):** Each grid cell have a Classification Loss that is the squared error of the conditional probabilities for each predefined class on YOLO:

$$\mathcal{L}_{classif} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \tag{3.6}$$

2. **Localization Loss ($\mathcal{L}_{loc}$):** It measures the errors between the location and size of

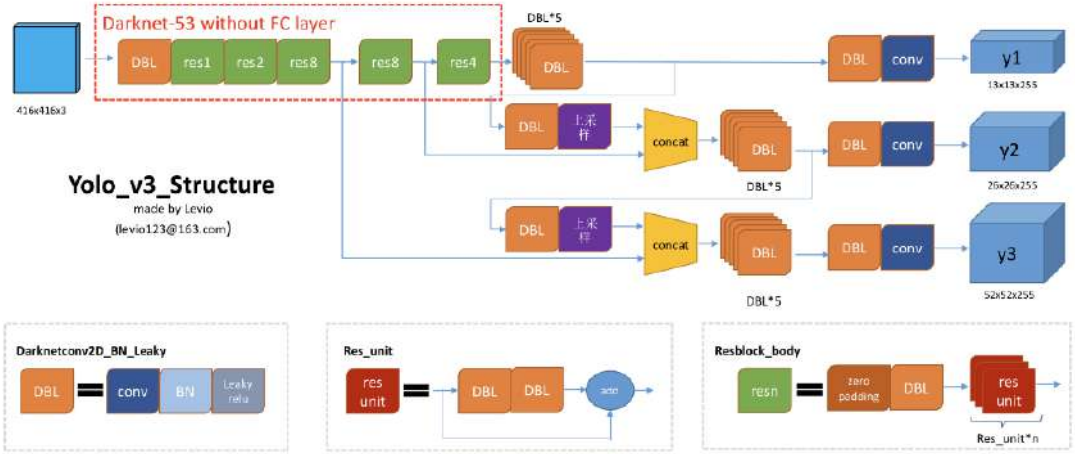


Figure 8: **YOLOv3 architecture.** Network Architecture of YOLO version 3.

each responsible bounding box for detecting the object:

$$\begin{aligned}\mathcal{L}_{loc} =& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right] \\ &+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right]\end{aligned} \tag{3.7}$$

3. **Confidence Loss ($\mathcal{L}_{conf}$):** it measures the probability that an object exists in each grid cell:

$$\mathcal{L}_{conf} = \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{ classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \tag{3.8}$$

In the above equations, each grid cell is presented with the index i and grid i is defined by (xi,yi),(wi,hi). Both wi and hi represents width and height, respectively, in relation to image size. The confidence score is represented by C_i, $\mathbb{1}_{ij}^{\text{obj}}$ represents if an object appears or not in cell i (i.e, takes 1 or 0) and $\mathbb{1}_{ij}^{\text{noobj}}$ is the complement of $\mathbb{1}_{ij}^{\text{obj}}$. And besides increasing the loss from the bounding box coordinate predictions and decreasing the loss from confidence for boxes that do not contain objects. Redmon et al. [29] uses two parameters. These parameters, λ_{coord} and λ_{noojb}, are both set up by default to 0.5.

The final loss $\mathcal{L}$ that YOLO uses during training is the sum of $\mathcal{L}_{classif}$, $\mathcal{L}_{loc}$ and $\mathcal{L}_{conf}$.

3.2.2 Test Details

At testing stage, the confidence score of each bounding box needs to be multiplied by conditional class probability, as we can see in Equation 3.9:

$$\begin{aligned} &\Pr(\text{ Object }) * IOU_{\text{pred}}^{\text{truth}} * \Pr(\text{ Class }_i \mid \text{ Object }) \\ &= \Pr(\text{Class}_i) * IOU_{\text{pred}}^{\text{truth}} \end{aligned} \tag{3.9}$$

$\Pr(\text{Class}_i) * IOU_{\text{pred}}^{\text{truth}}$ represents a class-specific confidence scores for each box. Redmon et al. [29] give us a good illustration to understand it at Figure 7.

3.3 Real-Time Recurrent Regression Networks

For our purpose, we focus on an algorithm called Re3 (Real-time, Recurrent, Regression-based) [15]. It is an accurate and efficient network for tracking generic objects in real-time.

Re3 is extremely fast and computationally cheap during inference due to shifting the computational burden offline. This shifting origin by the direct embedded in the network of the transformations caused by the change over time of the tracked object. In other words, Gordon, Farhadi, and Fox [15] trained Re3 to learn from many examples offline. Also, it quickly updates the appearance and motion models online when tracking a specific object and improves temporary occlusion compared to other trackers.

In detail, this network was trained over ILSVRC 2016 [58] Object detection from Video Dataset (Imagenet Video) and the Amsterdam Library of Ordinary Videos (ALOV) 300++ dataset [59]. It was proposed in Tensorflow over CaffeNet's pre-trained weights for its convolutional layers, which have skip-connection layers after layers: norm1, norm2, conv5. Each of them with 16, 32, and 64 channels, respectively. Also, each skip layer has a PReLu nonlinearity and the embedding fully-connected layer has 2048 units, and finally, the LSTM (Long Short-Term Memory) layers have 1024 units each.

The formulation in Equations (3.10 $\sim$ 3.15) shows how the two-layer factored LSTM with peephole connections is able to capture complex object transformations and remember longer-term relationships.

$$z^t = h\left(\mathbf{W}_z x^t + \mathbf{R}_z y^{t-1} + b_z\right) \tag{3.10}$$

$$i^t = \sigma\left(\mathbf{W}_i x^t + \mathbf{R}_i y^{t-1} + \mathbf{P}_i c^{t-1} + b_i\right) \tag{3.11}$$

$$f^t = \sigma\left(\mathbf{W}_f x^t + \mathbf{R}_f y^{t-1} + \mathbf{P}_f c^{t-1} + b_f\right) \tag{3.12}$$

$$c^t = i^t \odot z^t + f^t \odot c^{t-1} \tag{3.13}$$

$$o^t = \sigma\left(\mathbf{W}_o x^t + \mathbf{R}_o y^{t-1} + \mathbf{P}_o c^t + b_o\right) \tag{3.14}$$

$$y^t = o^t \odot h\left(c^t\right) \tag{3.15}$$

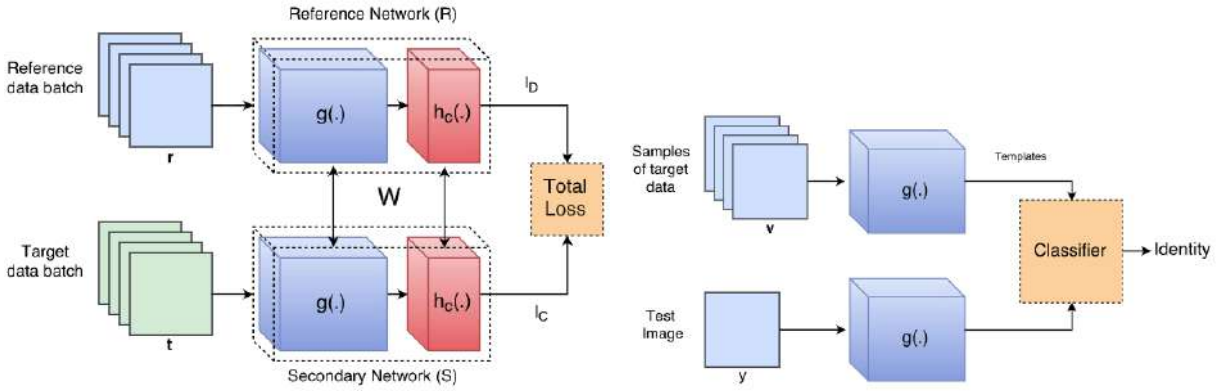
Where t represents the frame index, x^t is the current input vector, y^{t-1} is the previous output (or recurrent) vector, b is the bias vector, h is the hyperbolic tangent function, σ is the sigmoid function, and $\odot$ is point-wise multiplication. Also, Gordon, Farhadi, and Fox [15] described weight matrices for the input as $\boldsymbol{W}$, and described recurrent and peephole connections as $\boldsymbol{R}$ and $\boldsymbol{P}$ respectively. Both output vector(y^t) and cell state(c^t) are producted by forward pass. Output vector is used to regress the current coordinates, and the cell state holds important memory information. Finally, y^t and c^t are fed into the subsequent forward pass to propagate forward in time.

So, this strategy allows convolutional layers to embed the object appearance, recurrent layers to recall the appearance and motion of the object information, and a regression layer to output the object location. In other words, Re3 requires a bounding box around any object at time T. After that, Re3 produces bounding boxes for the Object in consecutive frames.

3.4 Learning Deep Features for One-Class Classification

The majority of Deep Learning models are susceptible to non-native class training data that it is resulting in incorrect class predictions [43]. In the context of this paper, we called a non-native class all classes that were not used for training. Thus, the non-native class may be called a novel or abnormal class detection, depending on the application domain [17].

Learning Deep Features for One-Class Classification (DOC) [17] takes advantage of the feature extraction network of CNN. Each feature vector that was extracted from input data represents and keeps a low intra-class variance that is embedded in a feature space



(a) (b)

Figure 9: (a)Training and (b)Testing architecture of DOC. Figure adapted from [17].

for the given class. For that purpose, Perera and Patel [17] introduced two loss functions are used to assess the quality of the learned deep feature, are called compactness loss and descriptiveness loss which work together into a parallel CNN model.

Also, DOC uses a compactness-loss to evaluate the closeness (compactness) of the native class among the learned space features, e.i. Different images of the same class have to contain a similar feature representation, and it has to allow finding a lower intra-class distance. Moreover, Perera and Patel [17] uses an external non-native multi-class dataset to evaluate a descriptiveness-loss where its objective is to find large intra-class distances [1].

Perera and Patel [17] highlight that these two important characteristics of features for one-class classification, compactness and descriptiveness, must be satisfied collectively thus making it possible to learn a more effective representation to achieve a useful feature. Formally, this optimization objective is by Perera and Patel [17] stated as follows,

$$\hat{g} = \min_{g} l_D(r) + \lambda l_C(t) \tag{3.16}$$

where l_C is compactness loss and l_D is descriptiveness loss, and r and t are the training data corresponding to the reference dataset, and to the given class, respectively, and λ is a positive constant.

In addition, Perera and Patel [17] consider the general case of the supervised deep learning-based classification model split into two main parts. A feature extraction (g) network and classifier (h_c) network. Furthermore, g is divided into feature-shared networks (g_s) and learned features networks (g_l).

[1] One objective of multiple-class is maximizing inter-class and minimizing intra-class distances [17].

In Figure 9 depict DOC proposes two architectures. (a) In training, we use both target and reference datasets, which are in fed into the network simultaneously. We thus need to consider two losses, compactness and descriptive loss. Also, Perera and Patel [17] start their formulation from a pre-trained model. For their purpose during training, they freeze the (g_s) network, and the (g_l) as well as (h_c) network learns. (b) During testing phase, we only use the sub-network g to obtain feature extraction.

Finally, DOC initializes two networks with identical weights to aim compactness and descriptiveness loss are evaluated based on the output of each network. In other words, Perera and Patel [17] use two kinds of image batches (from native class and non-native class datasets) for fed networks, respectively, where the native class network is predominance.

3.5 SiamIDL

To conduct our experiments, we perform classic Re-ID using the same Siamese neural network by Sumari et al. [9] called SiamIDL. The architecture of SiamIDL is composed as follows: layers of tied convolution with max-pooling (in which weights are shared across the two views), cross-input Neighborhood Differences (computes differences in features values), patch summary convolutional image features (to create a holistic representation of neighborhood difference maps), across-patch features, higher-order relationships (to learn relationships across neighborhood differences), and finally, a softmax function to estimate (score of similarity) whether the two input images are of the same person or not [51]

1. **Tied Convolution:** The first two layers of the network are convolution layers that are used to compute, on each input image, the higher-order features separately. The layers perform tied convolution in order for the features to be comparable across the two images in later layers. The shared weights across the two views ensure that the same filters are used to compute features. As shown in Figure 10, the first input pairs of RGB images are passed to the first convolution layer through learned filters, resulting in feature maps being passed through a max-pooling kernel; this halves the width and height features. Finally, these features are passed through another tied convolution layer, followed by a max-pooling layer that again decreases the width and height of the features by a factor of 2. As a result, each input image represented by 25 feature maps is obtained.

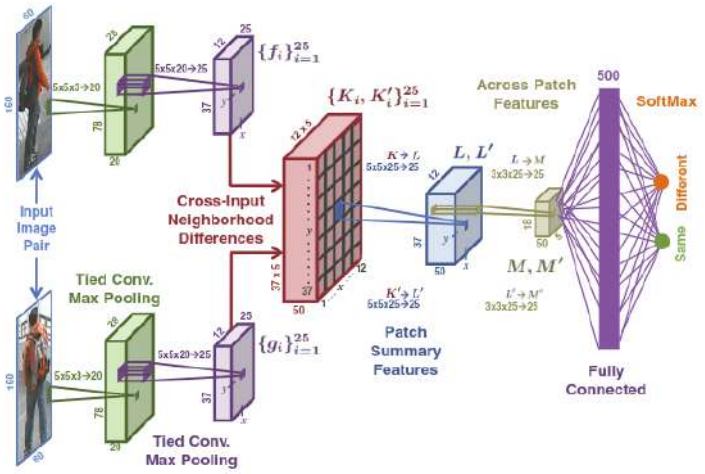


Figure 10: SiamIDL architecture. Image adapted from [51].

2. **Cross-Input Neighborhood Differences:** For each input image, a set of 25 feature maps is obtained from the two tied convolution layers, in which is possible to learn relationships between the two views. A cross-input neighborhood difference layer produces a set of 25 neighborhood difference maps by computing the differences in feature values in the two views around a neighborhood of each feature location,

3. **Patch Summary Features:** A patch summary layer summarizes the neighborhood difference maps (these maps express the rough relationship among features from the two input images) into a holistic representation of the differences of each block.

4. **Across-Patch Features:** The feature outcomes of the learn spatial relationships across neighborhood differences are computed by convolution layer with 25 filters. Next, these features are passed through a max pooling kernel to reduce the height and width by a factor of 2. Similarly, is obtained across-patch features.

5. Finally, a **fully connected layer** is applied to capture the higher-order relationships through: a) combining information from patches that are far from each other and b) combining information from both features. The 500 outputs, the resultant feature vector, are passed through a ReLu non-linearity, which contains 2 softmax units that represent the probability that the two images of the pair are of the same person or of different people.

Also, we use the same implementation by Sumari et al. [9], which used CUHK-03 dataset [60] as a training set containing 7239 images.

3.6 Bag of Tricks

Recently Luo et al. [50] proposed The Bag of Tricks approach. It has been collected from the observation of effective training tricks. This approach according resulted from the observation that most 370 improvements for Re-ID baselines come from neural network training tricks rather than Re-ID approaches themselves. As a result, Luo et al. [50] design a strong baseline for person Re-ID. They sum up simple sheep tips to achieve a successful train of the standard Re-Id model. The backbone used in our training was ResNet-50 [61].

We use the following parameters to conduct our experiments here. It was initialized the ResNet-50 model with pre-trained parameters on ImageNet, *softmax weight=1.0*, triplet loss weight=1.0, center *loss weight=0.0005*, and finally, Adam optimization was adopted. A train was done with the Market-1501 dataset with *epochs=200*, *batch size=64*, and images were resized to $256x128$ pixels. Market-1501 contains 12937 images to train and 19733 images to test.

4 TrADe Re-ID Methodology

In this chapter, we introduce a novel Live Re-ID approach called TrADe, which intends to improve the performance of practical Re-ID applications. This section presents the different components of our proposed method and further information about the TrADe implemented pipeline, i.e., pipeline, dataset, metrics used, and experiments conducted.

4.1 Overview of the Approach

A Live Re-ID pipeline receives as input a short video sequence and a query image. It must return whether the query is present in the video, as well as information regarding where and when it appears, as we illustrate in Figure 5. The baseline Live Re-ID pipeline proposed by Sumari et al. [9] uses an object detection model (YOLOv3) to locate pedestrians in every frame of the video.

The bounding boxes predicted by the model are used to build a search gallery, in which the query is sought. To do this, a classic Re-ID model is used (SiamIDL by Ahmed, Jones, and Marks [51]). The issue with this approach is that it generates very large galleries, containing some very bad images due to errors of the object detector. Search gallery is graphically depicted in Figure 11, where baseline Live Re-ID was applied to a 2 minutes segment video from the PRID [13] dataset when is applied the baseline Live Re-ID. Also, we highlight both correct and wrong cropping bounding boxes that are represented with white and red colors, respectively. Thus, we can see 764 red-bad bounding boxes (errors) and 3,507 well-white bounding boxes.

Here, we introduce an approach called TrADe, which is able to reduce the gallery size and improve the quality of its images. It relies on using a Tracking algorithm to identify bounding boxes representing the same individual in consecutive frames.Such a sequence of bounding boxes is called a tracklet. Then, an Anomaly Detection algorithm is used to select a single good image to represent each of these tracklets. This process generates smaller search galleries, containing images of better quality. Lastly, a classic Re-ID model

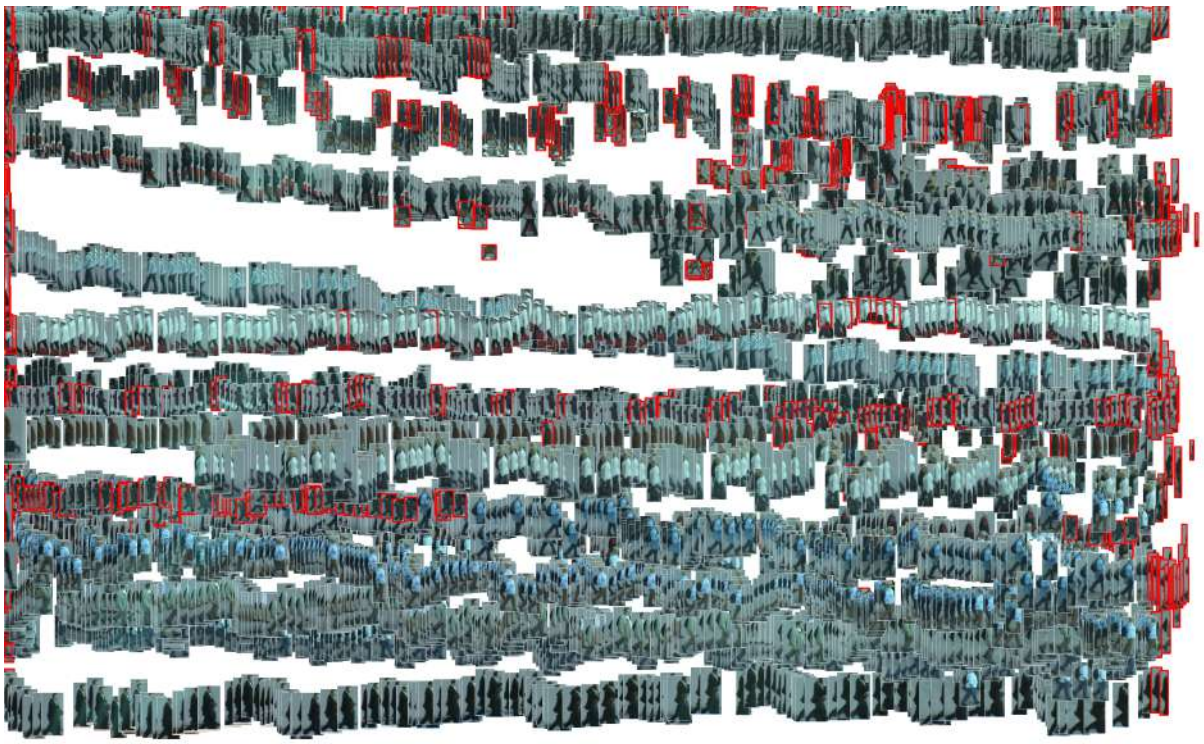


Figure 11: Errors of search gallery in baseline Live Re-ID.

is applied to the images of the gallery to compute their corresponding similarity scores with respect to the query image.

In Figure 12. TrADe repeats its process every N video frames input, i.e., over the first frame TrADe applies an object detection algorithm with the intention of achieving the initial bounding box (bounding boxes are highlighted), and subsequent $N-1$ TrADe applies a tracking detection algorithm in order to identify the same person in $N-1$ consecutive frames (bounding boxes are colorless). Before started again a TrADe's actions, it saves N cropping outcomes as a short track called tracklet with a global identifier.

This figure illustrates the general view of our proposal: (1) and (2) compose the gallery generation module, and (3) is the classic re-identification module of Live Re-ID.

Figure 13 proper a general overview of TrADe proposal. In the same manner, as baseline Live Re-ID uses two modules that are composed of two main modules the gallery generator and the classic Re-ID module. TrADe takes the same address, (1) and (2) encapsulates the gallery generator module, and (3) classic Re-ID module. Also, the whole pipeline uses a raw video and a query image as inputs and returns the list of the most similar detections and their corresponding scores, which are used to decide whether the query is present in the video. Orange shapes indicate that deep neural networks are used

Figure 12: Initial bounding boxes for each tracklet. We illustrate an example of a complete track extracted from 25 consecutive frames where the person's path inits on the bottom-right to go up on the top-left.

for the modules.

The gallery generator module is composed of two submodules (1) the frames of video stream are fed to the object detector, to find the initials bounding boxes that feed follow step. After, initials bounding boxes are fed to object tracking to generate tracklets of maximum length N. (2) Anomaly Detector feeds with tracklets generated in (1) in order to find the best candidate for each of them, thus forming a search gallery. In the classic Re-ID module (3), the query image of a person is searched in the gallery previously generated, which outputs a list of images similar to the query, shorted from most similar to least similar.

4.2 Generating the Tracklets

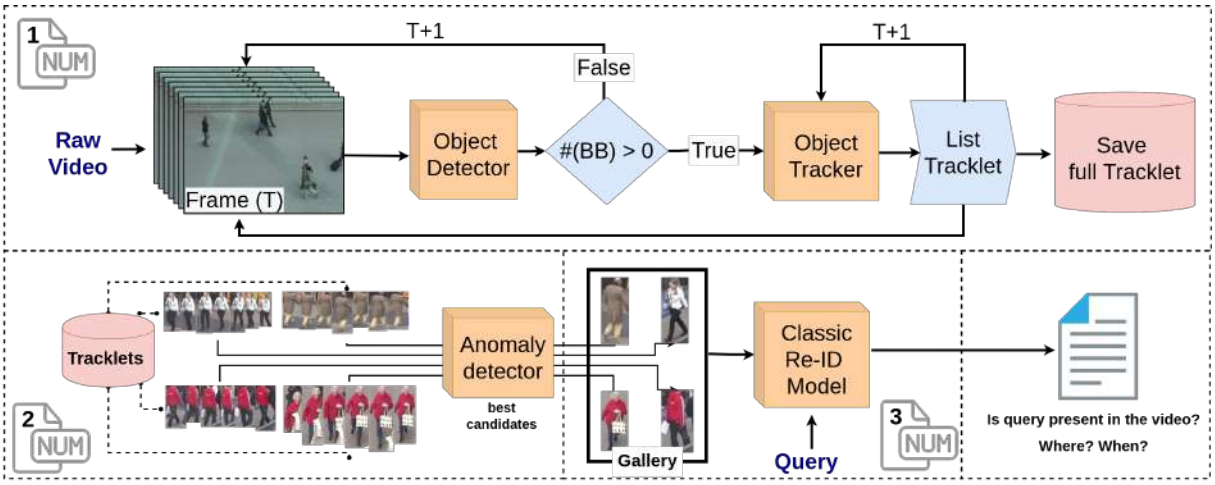


Figure 13: Overview of TrADe Live Re-ID.

The first step in TrADe consists in generating short tracklets of consecutive bounding

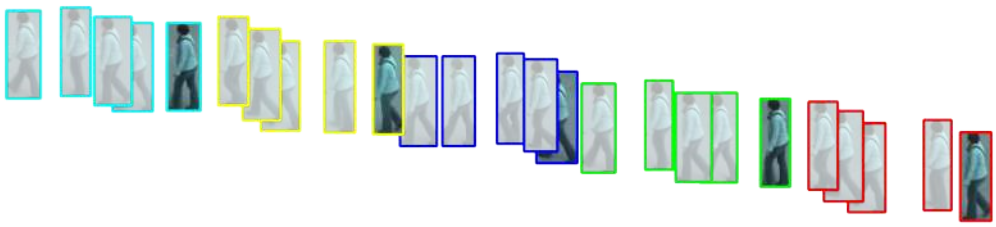


Figure 14: Generate tracklets.

boxes representing the same individual. To do this, the first frame of the search video is processed by the object detection model in order to generate initial bounding boxes. If no pedestrian is detected in this frame, we keep searching in the following frames until we have at least one initial bounding box. Then, the detected bounding boxes are provided to the object tracking model for initialization. The tracker keeps running in the following frames to generate a sequence of consecutive bounding boxes, called a tracklet.

In order to illustrate this, we present Figure 14 and as we mentioned above TrADe applies an object detection (YOLOv3) algorithm with the intention of achieving the initial bounding box, we then use these bounding boxes to input TrADe's object tracking (Re3) algorithm on the following $N-1$ consecutive frames. Besides, we highlight initial bounding boxes at beginning of each tracklet, and each tracklet is represented by same color box.

One of the main issues with modern tracking algorithms is known as the label-switching problem. It happens when people cross (bounding boxes overlap), or when one goes out of the frame and another enters a few frames later at a nearby location. This can lead to very long tracklets, containing different persons, which is an undesirable property for TrADe. Indeed, as TrADe only selects a single bounding box to represent an entire tracklet, if several persons appear in the same tracklet, some might not be represented in the final gallery. To address this issue, we force TrADe to generate small tracklets by fixing their maximum size. We define as N the maximum tracklet length, which is a user-defined parameter. In practice, whenever a tracklet contains N frames, it is stopped. Then, the object detection model is run every N frames to initialize new tracklets, representing people who entered the camera field of view during the video. The influence of the parameter N on Live Re-ID results is evaluated in our experiments.

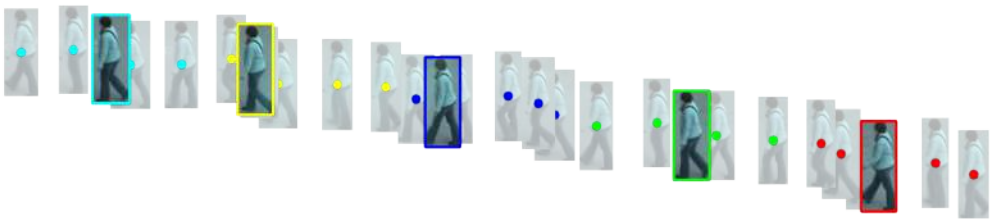


Figure 15: Choose the best candidate with DOC

4.3 Selecting a Single Image to Represent a Tracklet

Once short tracklets have been generated by the above module, we want to select a single good image for each tracklet to enter the search gallery. A good image is defined as a properly cropped image containing the entire body of a single human being. By doing this, we aim to remove badly cropped images that were shown to decrease Live Re-ID performance by [9]. In other words, we want to generate galleries that contain images belonging to the same domain as most classic Re-ID training datasets.

To select the representative image for a tracklet, we use an anomaly detection approach called DOC [17]. It is a one-class classifier that is trained to distinguish good images for Re-ID from bad ones. The DOC anomaly detector is composed of two parts, a feature extraction network that produces a representation adapted to identify bad human images, and a classifier network, which produces a score representing the "goodness" of the input image. This score is then used to select the best image of the tracklet, i.e., the one with the highest score.

In Figure 15 clearly shows the outcome over a simple trajectory from consecutive frames of video segments. We depicted, after once having tracklets generated, to apply TrADe's anomaly detection (DOC) algorithm of each tracklet. We highlight best bounding boxes chosen by DOC at each tracklet, and everyone else in each tracklet is represented by same color box.

4.4 Practical Implementation Choices

Here, we present the practical implementation details to reproduce our results. The complete code is available on GitHub `https://github.com/luigy-mach/TrADe`.

4.4.1 Pedestrian Detection

For our proposes, we prepared the model with the following settings:

1. The network only uses a class corresponding to the person (pedestrians) and
2. A limited threshold of Intersection Over Union.

Intersection Over Union (IoU) helps us benchmark the accuracy of the model predictions. During our evaluation (Section 4), we established the IoU by 0.5 to generate bounding boxes of pedestrians in a specific frame, as we can see in Figure 4, scored above 50% considered the threshold for good detection.

Henceforth, we refer to the detection under specific application scenarios, i.e., pedestrian detection. Inside our TrADe pipeline, YOLOv3 acts like a detector pedestrian to initialize bounding boxes to lead pedestrian tracking. As we can see in Figure 13, YOLOv3 is called in three main cases.

1. When is the frame, $T = 0$, in the sequences of video,
2. Whether in the current frame, T, does not have any bounding box of a pedestrian.
3. If either some tracklet is full (N) or the iteration pipeline according to a frame (N) is complete.

We use the pre-trained version of YOLOv3 proposed in TensorFlow, which was trained on PASCAL VOC 2012 [27], containing 20 classes. The backbone for YOLOv3 is Darknet-53.

We prepared the model as follows:

1. the network only uses the "person" output class,
2. all bounding boxes with classification score below 0.5 are rejected.

4.4.2 Pedestrian Tracking

For tracking, we used the pre-trained Re3 model from the official repository[1]. It was trained using ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2016 [58]

[1] https://github.com/danielgordon10/re3-tensorflow

Object detection from Video Dataset (Imagenet Video) and the Amsterdam Library of Ordinary Videos 300++ [59]. It is a Tensorflow implementation over CaffeNet's pre-trained weights.

4.4.3 Anomaly Detection

To be able to train the DOC classifier, we need sufficient example images of the target class (good images for classic Re-ID), as well as non-target class (representing non-class person images). To build this DOC training dataset, we use a subset of the CUHK03 [60] Re-ID dataset as a class target, and the non-target class is collected from the VOC2012 dataset [27] (all classes except person class).

4.4.4 Classic Re-ID

We experimented TrADe with two different classic Re-ID approaches. The first one is SiamIDL [51], and we use the same implementation as [9], trained on CUHK-03 [60]. We also use a more recent approach called BoT (Bag of Tricks) [50]. The backbone used in our training was ResNet-50 [62]. We used ImageNet pre-trained weights for initialization and fine-tuned the BoT Re-ID model using the Market-1501 [63] dataset.

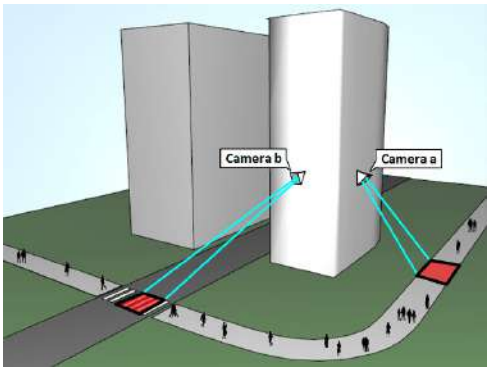
4.5 Experimental Evaluation

In this section, we describe the details of the experimental evaluations conducted in this work.

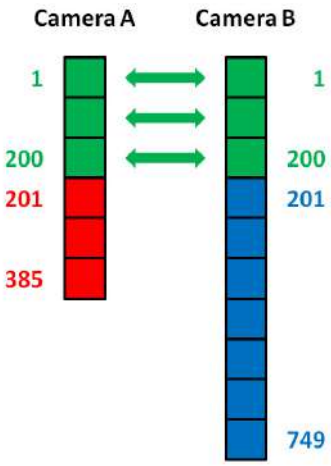
4.5.1 Dataset

To the best of our knowledge, today there is only one public dataset that can be used to evaluate complete Live Re-ID pipelines, which was introduced in [9]. It is a modified version of the PRID-2011 dataset [13], based on the raw video footage and the original annotations that were used to create the official version of PRID2011.

The PRID-2011 videos were collected from two non-overlapping cameras located in Graz, Austria. The live PRID dataset was extracted from two cameras (cam A and cam B). Cam A has a duration of *1h 01m 52s* and cam B has *1h 06m 39s*. Furthermore, It contains 385 identities for the first camera and 749 for the second, with 200 shared



(a) two different, static surveillance cameras (A&B) and location of they.



(b) Shared 200 IDs between cameras (A&B)

Figure 16: PRID 2011 dataset. Adapted from [13]

identities across both cameras. The modified PRID (live PRID) dataset contains several two minutes videos (63), and for each short video, it has a ground truth file associated with information about each individual it contains. For evaluation, we consider 73 queries in total.

In Figure 16 shows the scenarios where videos are taken to generate PRID2011 dataset. (a) The dataset consists of two videos recorded multiple-person trajectories recorded from two different cameras. Both cameras contain a viewpoint change and stark differences in illumination, background, and camera characteristics. (b) PRID2011 contains 385 identities for camera A and 749 for camera B, both have 200 shared identities, i.e., a person with ID 001 in camera A corresponds to a person with ID 001 in camera B.

To evaluate TrADe, we apply the same evaluation methodology as [9]. We select ten videos of two minutes from each camera. Between each pair of videos, we select the persons who appear in both cameras. Approximately the first four query images for each video were selected and exchanged between each video by ensuring that each query appears at least in one frame. In total, our evaluation consists of 20 videos and 73 queries.

Using the notations from [9], we use the following parameters for our Live Re-ID pipeline:

- The number of frames for video splitting (τ) is set to 1000, which was the best value from [9] experiments.
- For β, the threshold on Re-ID scores for generating an alert to the monitoring agent,

we use values between 0 and 1 with a step size of 0.02.

- The number of candidates shown to the monitoring agent (η) is set to 20, which was also the best value in [9].

4.5.2 Evaluation Metrics

We use the following evaluation metrics [9]:

- The Finding Rate (FR) is the proportion of short videos such that the query was present and presented to the monitoring agent. A low FR occurs when the query is missed frequently.
- The True Validation Rate (TVR) is the proportion of alerts raised to the agent such that the query was among the presented candidates. A low TVR occurs when the monitoring agent was frequently unjustified disturbed.

The next equations, 4.1 and 4.2, define the metrics FR and TVR, respectively:

$$FR = \frac{TC}{TC + TMC + FS} \tag{4.1}$$

$$TVR = \frac{TC}{TC + TMC + FC} \tag{4.2}$$

To better present the results, we use the following two metrics to ease the interpretation of TrADe Re-ID results:

- Similarly to the mean Average Precision (mAP) for standard object detection approaches, we compute the area under the *TVR vs FR* curve and call it mAP by analogy. This allows to present results that are independent of the threshold β.
- Similarly to the F-score computation, as shown in Equation 4.3, for precision and recall, we compute the F_1 score for FR and TVR as their harmonic mean.

$$F_\gamma = \left(1 + \gamma^2\right) \cdot \frac{\mathrm{FR} \cdot \mathrm{TVR}}{(\gamma^2 \cdot \mathrm{FR}) + \mathrm{TVR}} \tag{4.3}$$

However, each value of the threshold β involves a different value of F_1. To address this problem, we use the optimal configuration for F_1 and call it F_1^* (see [64]). In other words,

it corresponds to the highest value of F_1 among all possible values of β. An F_1^* of 1 means that there exists a Re-ID threshold β such that the Live Re-ID pipeline works perfectly. When single values of FR and TVR are reported, they are the ones corresponding to the optimal F_1 threshold. Equation 4.4 describe a F_1^* formula.

$$\begin{aligned} F^*_{\gamma=1} &= \left(1+1^2\right) \cdot \frac{\mathrm{FR} \cdot \mathrm{TVR}}{(1^2 \cdot \mathrm{FR}) + \mathrm{TVR}} \\ &= 2 \cdot \frac{\cdot \mathrm{FR} \cdot \mathrm{TVR}}{\mathrm{FR} + \mathrm{TVR}} \end{aligned} \tag{4.4}$$

4.5.3 Comparison with Other Approaches

Our TrADe Re-ID approach for Live Re-ID is compared against the baseline presented in [9], which corresponds to the limit case when the maximum length of the tracklet (N) is set to one. To evaluate if the benefits of TrADe are only due to the reduced gallery size, we also compare TrADe against a simpler approach for gallery size reduction, which we call *Skip*. It consists in simply running the YOLOv3 object detector once every N frames, where N is the maximum tracklet size for TrADe. This simple approach generates galleries of the same size as TrADe and allows us to evaluate the impact of the anomaly detection component of TrADe.

5 Results

In this chapter, we present and discuss the results obtained with TrADe. To evaluate TrADe method we use live-PRID dataset (see Section 4.5.1) proposed in order to validate the effectiveness of our new strategy for Live Re-ID deployment. The evaluation process includes F-score, and mean Average Precision (mAP) metrics (described in Chapter 4). Furthermore, we display the results over live-PRID dataset in order to compare both TrADe and classic Live Re-ID proposed by [9].

For experimental results, we test all values of N between 1 to 80 with a multiplication step size of 2, where the immediate value after 1 is 5, i.e., $N \in \{1,5,10,20,40,80,...\}$. In addition, we use a Skip notation to reference to Live Re-ID that was applied to every N frame in order to compare with our proposal(see Section 4.5.3). And finally, we use two different Classic Re-ID models in order to reach our assumptions. We use two classic Re-ID models, SiamIDL and BoT.

Our results demonstrate that the TrADe proposal represents a significant improvement in state-of-art and is sufficient to draw our conclusions.

5.1 Qualitative Observations

Figure 20 clearly illustrates the process of TrADe methodology. TrADe aims to improve the performance of Live Re-ID systems, and intuitively reduce the time processing in classic Re-ID module, because performs a drastic reduction of search gallery. We observe in the contrast between Figures 20(a & b).

So, Figure 20(a) denoted massive bounding boxes from a short video segment, that was observed in baseline Live Re-ID, which takes more effort and delays to classic Re-ID module. Also, this approach is prone to errors as the generated galleries are very large and contain a large quantity of poorly cropped human bodies. We illustrated these bad bounding boxes in Figure 11 where highlight error with red boxes.

Also, Figure 20(b) describes the outcomes of object detector and object tracking in order to create tracklets. It is a core behind TrADe approach because we need to group (e.i, link between sequence frames) bounding boxes to connect short tracks (tracklets), i.e., consecutive boxes displaying the same person, represented by adjacent boxes with the same color. And finally, Figure 20(c) shows a dramatically reduced search gallery because TrADe uses an Anomaly Detector to select a single, good, and representative bounding box of each tracklets and, consequently, improve the quality and reduce the number of items in search gallery.

5.2 Performance of TrADe

		FR	TVR	F_1^*	mAP
SiamIDL	Baseline [9]	0.544	0.196	0.289	0.104
	Skip	0.792	**0.500**	0.422	0.258
	TrADe	**0.823**	**0.500**	**0.439**	**0.279**
BoT	Baseline [9]	0.506	0.188	0.268	0.095
	Skip	0.835	**0.387**	0.463	0.302
	TrADe	**0.886**	0.372	**0.481**	**0.317**

Table 1: **live-PRID results**. Results obtained with different Live Re-ID approaches (including TrADe) on the live-PRID dataset. These results are for $N = 20$.

The results obtained with different approaches are reported in Table 1. The results presented are for $N = 20$, which produces a good trade-off between gallery size reduction and loss of information.

We can see that for both classic Re-ID approaches (SiamIDL and BoT), it is generally a good idea to reduce the gallery size. Indeed both the simple approach (Skip) and TrADe perform significantly better than the baseline from [9]. This means that galleries generated by simply using the object detector on every frame are too large to be processed correctly by the classic Re-ID models and generate noise.

We can also see that TrADe performs almost always better than Skip. This means that using the Anomaly Detection module for selecting good images to represent tracklets is a good idea for Live Re-ID. Overall, these results confirm that TrADe is a promising approach to address the Live Re-ID problem, leading to significant improvements over the current state-of-the-art baseline proposed in [9].

5.3 Influence of the Maximum Tracklet Size N

Method	**cam A$_{\#BB}$**	**cam B$_{\#BB}$**
Skip	138190	96559
TrADe $_{N=5}$	27849	20102
TrADe $_{N=10}$	14006	10299
TrADe $_{N=20}$	7094	5315
TrADe $_{N=40}$	3610	2371
TrADe $_{N=80}$	**1935**	**1536**

Table 2: Influence of N in size of search gallery. We use raw videos of PRID [13] dataset. Video length in camera A is *1:01:52* hours and in camera B during *1:06:39* hours..

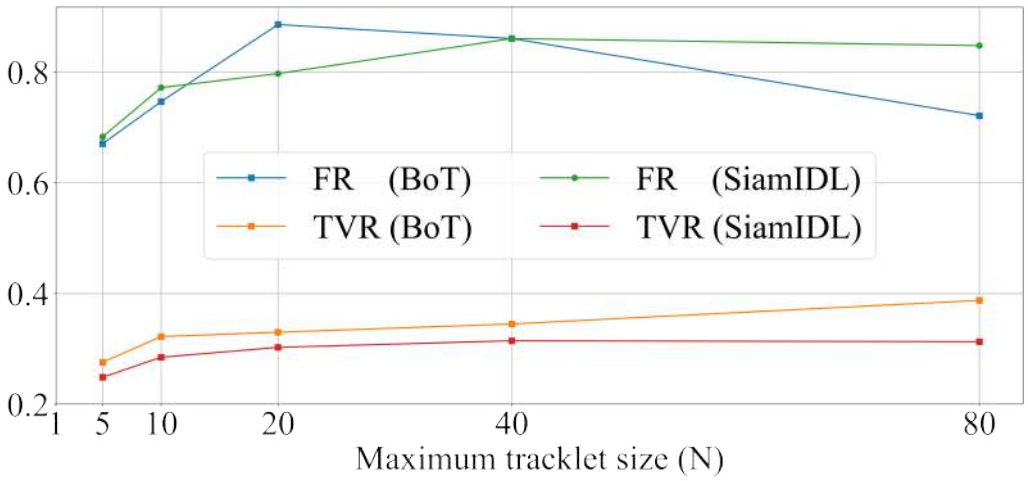


Figure 17: Influence of N on mAP and F_1^*. These graphs show the values taken by two important metrics (mAP and F_1^*) for different values of the hyperparameter N. We do not forget that TrADe methodology sets value $N = 1$ to represent a Skip methodology.

Here, we discuss the influence of the hyperparameter N, which is the maximum length of a tracklet on the Live Re-ID results obtained with TrADe. We test different values of N on a log-scale: $N \in \{1,5,10,20,40,80,...\}$, in both classic Re-ID algorithms(i.e., SiamIDl and BoT), and the curves representing the evolution of F_1^*, mAP, and the time required to run classic Re-ID on the generated gallery. For this purpose, both raw videos of Live PRID are fed to TrADe in order to generate search galleries We must not forget that TrADe methodology sets value $N = 1$ to represent a Skip methodology.

Figure 17 shows that increasing N helps to improve the TVR (True Validation Rate). This makes sense as larger values of N lead to smaller galleries, containing less misleading images, which in turn generate fewer false alarms. On the other hand, we can see that this

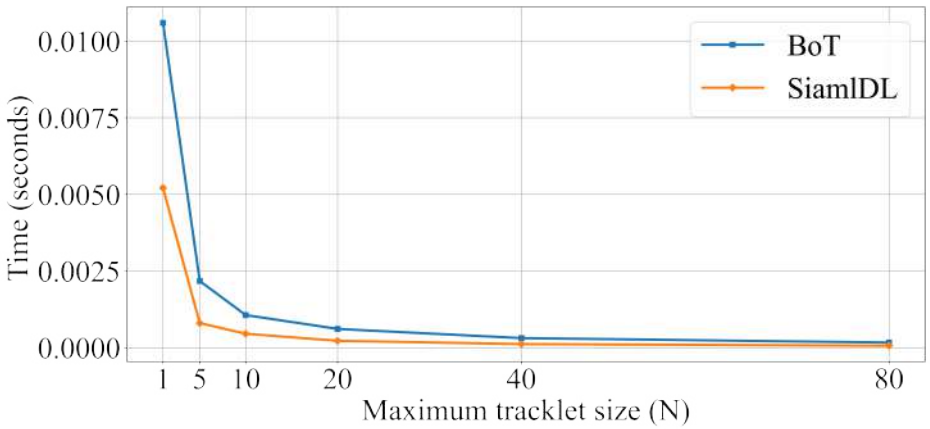


Figure 18: Influence of N in time of reidentify.

pattern is less clear for the FR (Finding Rate), which starts decreasing for BoT when N is above 20. This also makes sense, because when we allow very long tracklets, some persons will not appear in the final gallery due to the label switching problem (see Section 4.2). For these reasons, we used a value of $N = 20$ in the experiments of Section 5.2.

The second result is that the classic Re-ID processing time appears to decrease drastically as we increase the size of the tracklet N (Figure 18). This makes intuitive sense, as when N increases, the size of the gallery decreases, and by extension the classic Re-ID module needs to process fewer images. We also note that the time decreasing effect is less pronounced when N exceeds 20. This is because most images in the gallery can already fit in a single batch for GPU processing at that point.

In order to highlight a drastic reduction size of the search gallery we present a Table 2. For our purpose, we use the two raw videos of Live PRID that were extracted from two cameras (A&B). Video length in camera A is *1:01:52* hours and in camera B during *1:06:39* hours. In this Table, we observe that as the value of N increases, the size of the search gallery decreases, which means that the classic Re-ID module has to process fewer images. This relationship makes intuitive sense, making us think that the best value is the highest of N. However, as shown in Figure 19, the Average precision does not increase in the same proportion of N.

Additionally, we present a curve of Average Precision in the same manner as our previous work [9] in order to evaluate whole performance pipeline of Live Re-ID. So, the influence in N becomes an appropriate indicator for emphasizing in the improvement of AP metric, as we can see in Figure 19.

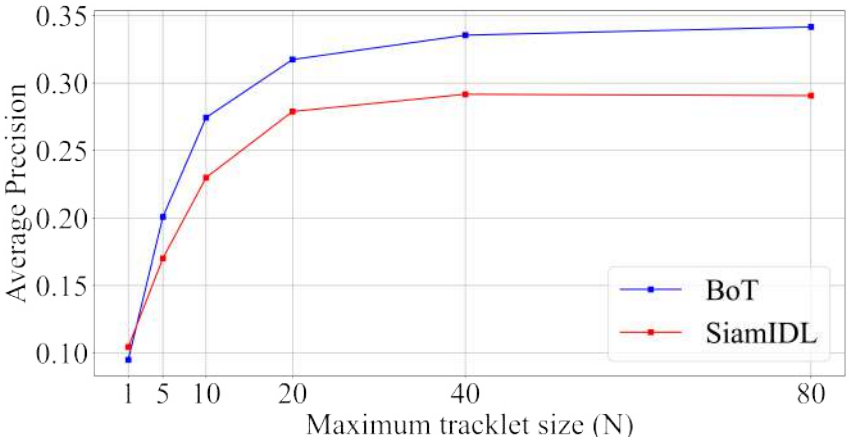
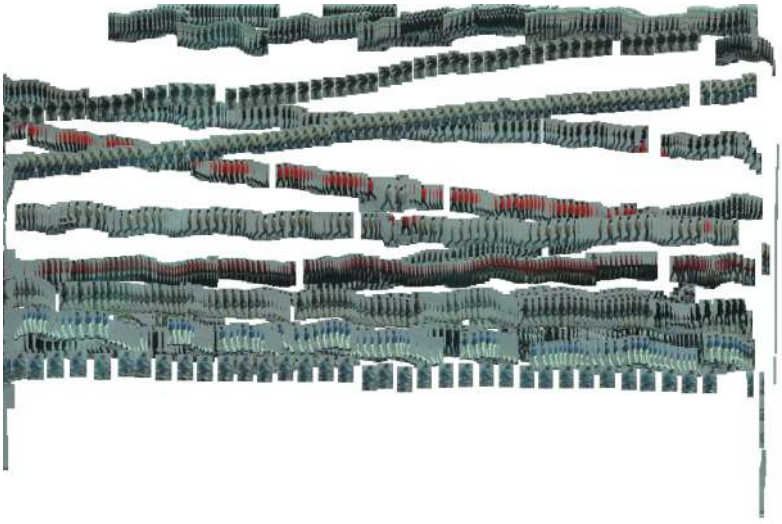
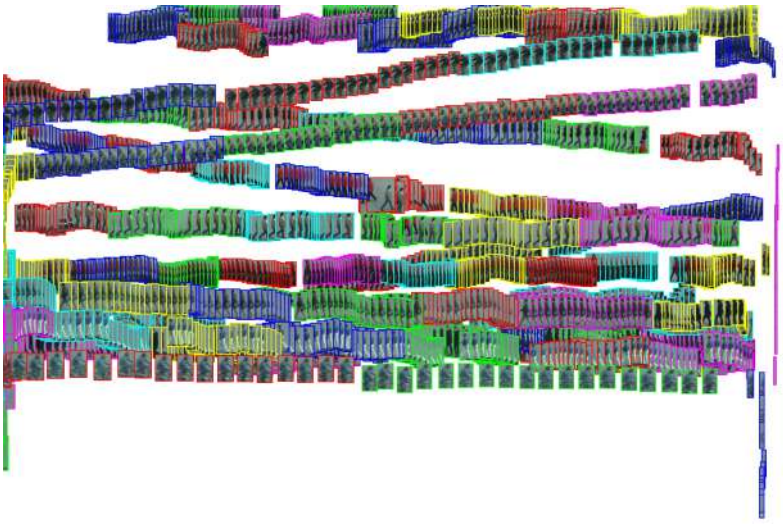


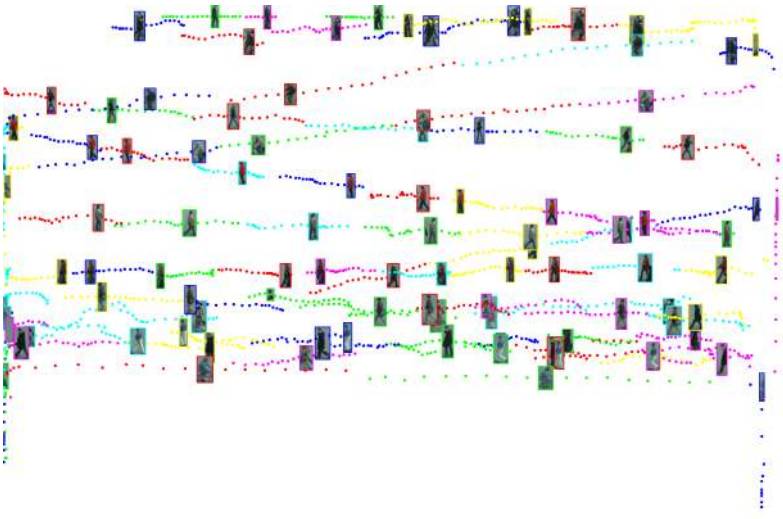
Figure 19: Influence of N in Average precision of TrADe pipeline.



(a) YOLOv3 gallery



(b) After Tracking



(c) TrADe gallery

Figure 20: Intuition behind TrADe Live Re-ID

6 Conclusions and Future Works

In this work we addressed the Live Re-ID problem, which uses raw videos as search galleries instead of manually cropped full-body images. A first baseline approach for Live Re-ID was proposed in our previous work [9], using object detection to generate a search gallery, and classic Re-ID to find the query in the gallery. A major issue with this baseline is the fact that the galleries obtained are too large, and contain outlier images, which do not represent human bodies.

Our proposal is to use a tracking algorithm to identify when successive bounding boxes are of the same individual, and group them as tracklets. Following, an anomaly detection model is used to select the "most normal" image of each tracklet. This approach is called TrADe and generates lower galleries than the baseline, with fewer outliers. Our experimental results confirm that TrADe performs much better than the baseline, TrADe is a significant step toward building better Re-ID applications.

We also present two ideas that could be explored in the future. First, our pipeline uses several deep neural networks that were all pre-trained on ImageNet (initial layers). Hence, we could speed up the pipeline considerably by building a single architecture and sharing the generic first layers of the neural network, used among the four modules based on deep learning. A second promising idea would be to consider not only a single image per tracklet, but rather several good quality images that are complementary, i.e., representing different poses of a person. The good results obtained recently on the video-based Re-ID setting [7] suggest that this could have a positive impact on Live Re-ID results.

References

[1] Julian Laufs, Hervé Borrion, and Ben Bradford. "Security and the smart city: A systematic review". In: *Sustainable Cities and Society* 55 (2020), p. 102023. ISSN: 2210-6707. DOI: https://doi.org/10.1016/j.scs.2020.102023.

[2] Dimitrios N. Serpanos and Andreas Papalambrou. "Security and Privacy in Distributed Smart Cameras". In: *Proceedings of the IEEE* 96.10 (2008), pp. 1678–1687. DOI: 10.1109/JPROC.2008.928763.

[3] Yundong Guo et al. "Multi-person multi-camera tracking for live stream videos based on improved motion model and matching cascade". In: *Neurocomputing* 492 (2022), pp. 561–571. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.12.047.

[4] A. Hampapur et al. "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking". In: *IEEE Signal Processing Magazine* 22.2 (2005), pp. 38–51. DOI: 10.1109/MSP.2005.1406476.

[5] A. Hampapur et al. "Smart surveillance: applications, technologies and implications". In: *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*. Vol. 2. 2003, 1133–1138 vol.2. DOI: 10.1109/ICICS.2003.1292637.

[6] Yi-Ling Chen et al. "Intelligent Urban Video Surveillance System for Automatic Vehicle Detection and Tracking in Clouds". In: *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. 2013, pp. 814–821. DOI: 10.1109/AINA.2013.23.

[7] Xiujun Shu et al. "Diverse part attentive network for video-based person re-identification". In: *Pattern Recognition Letters* 149 (2021), pp. 17–23.

[8] Mang Ye et al. "Deep Learning for Person Re-Identification: A Survey and Outlook". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.6 (2022), pp. 2872–2893. DOI: 10.1109/TPAMI.2021.3054775.

[9] Felix O. Sumari et al. "Towards practical implementations of person re-identification from full video frames". In: *Pattern Recognition Letters* 138 (2020), pp. 513–519. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2020.08.023.

[10] Apurva Bedagkar-Gala and Shishir K. Shah. "A survey of approaches and trends in person re-identification". In: *Image and Vision Computing* 32.4 (2014), pp. 270–286. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2014.02.001.

[11] Damla Gül Altunay et al. "Intelligent surveillance system for abandoned luggage". In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. 2018, pp. 1–4. DOI: 10.1109/SIU.2018.8404327.

[12] Debayan Deb, Divyansh Aggarwal, and Anil K Jain. "Identifying Missing Children: Face Age-Progression via Deep Feature Aging". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10540–10547.

[13] Martin Hirzer et al. "Person Re-identification by Descriptive and Discriminative Classification". In: *Image Analysis*. Ed. by Anders Heyden and Fredrik Kahl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 91–102. ISBN: 978-3-642-21227-7.

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

[15] Daniel Gordon, Ali Farhadi, and Dieter Fox. "Re^3: Re al-Time Recurrent Regression Networks for Visual Tracking of Generic Objects". In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 788–795. DOI: 10.1109/LRA.2018.2792152.

[16] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: *ArXiv* abs/1804.02767 (2018).

[17] Pramuditha Perera and Vishal M. Patel. "Learning Deep Features for One-Class Classification". In: *IEEE Transactions on Image Processing* 28.11 (2019), pp. 5450–5463. DOI: 10.1109/TIP.2019.2917862.

[18] Luigy Machaca et al. "TrADe Re-ID–Live Person Re-Identification using Tracking and Anomaly Detection". In: *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2022. DOI: https://doi.org/10.48550/arXiv.2209.06452.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

[20] Li Deng. "A tutorial survey of architectures, algorithms, and applications for deep learning". In: *APSIPA transactions on Signal and Information Processing* 3 (2014).

[21] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, pp. 1–6.

[22] Yanming Guo et al. "Deep learning for visual understanding: A review". In: *Neurocomputing* 187 (2016), pp. 27–48.

[23] Junyi Chai et al. "Deep learning in computer vision: A critical review of emerging techniques and application scenarios". In: *Machine Learning with Applications* 6 (2021), p. 100134.

[24] Li Liu et al. "Deep Learning for Generic Object Detection: A Survey". In: *Int. J. Comput. Vision* 128.2 (Feb. 2020), pp. 261–318. ISSN: 0920-5691. DOI: 10.1007/s11263-019-01247-4.

[25] Licheng Jiao et al. "A Survey of Deep Learning-Based Object Detection". In: *IEEE Access* 7 (2019), pp. 128837–128868. DOI: 10.1109/ACCESS.2019.2939201.

[26] Ben Benfold and I. Reid. "Stable multi-target tracking in real-time surveillance video". In: *CVPR 2011* (2011), pp. 3457–3464.

[27] Mark Everingham et al. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[28] Wei Liu et al. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[29] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[30] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[31] Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6517–6525. DOI: 10.1109/CVPR.2017.690.

[32] Xi Li et al. "A Survey of Appearance Models in Visual Object Tracking". In: *ACM Trans. Intell. Syst. Technol.* 4.4 (Oct. 2013). ISSN: 2157-6904. DOI: 10.1145/2508037.2508039.

[33] Alper Yilmaz, Omar Javed, and Mubarak Shah. "Object Tracking: A Survey". In: 38.4 (Dec. 2006), 13–es. ISSN: 0360-0300. DOI: 10.1145/1177352.1177355.

[34] Mustansar Fiaz et al. "Handcrafted and Deep Trackers: Recent Visual Object Tracking Approaches and Trends". In: *ACM Comput. Surv.* 52.2 (Apr. 2019). ISSN: 0360-0300. DOI: 10.1145/3309665.

[35] Rabah Iguernaissi et al. "People tracking in multi-camera systems: a review". In: *Multimedia Tools and Applications* 78.8 (Apr. 2019), pp. 10773–10793. ISSN: 1573-7721. DOI: 10.1007/s11042-018-6638-5.

[36] Milan Ondrašovič and Peter Tarábek. "Siamese Visual Object Tracking: A Survey". In: *IEEE Access* 9 (2021), pp. 110149–110172. DOI: 10.1109/ACCESS.2021.3101988.

[37] Hanxuan Yang et al. "Recent advances and trends in visual tracking: A review". In: *Neurocomputing* 74.18 (2011), pp. 3823–3831. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2011.07.024.

[38] Mohammed Y. Abbass et al. "A survey on online learning for visual tracking". In: *The Visual Computer* 37.5 (May 2021), pp. 993–1014. ISSN: 1432-2315. DOI: 10.1007/s00371-020-01848-y.

[39] Wenhan Luo et al. "Multiple object tracking: A literature review". In: *Artificial Intelligence* 293 (2021), p. 103448. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2020.103448.

[40] Guansong Pang et al. "Deep Learning for Anomaly Detection: A Review". In: 54.2 (Mar. 2021). ISSN: 0360-0300. DOI: 10.1145/3439950.

[41] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Comput. Surv.* 41.3 (July 2009). ISSN: 0360-0300. DOI: 10.1145/1541880.1541882.

[42] Raghavendra Chalapathy and Sanjay Chawla. "Deep learning for anomaly detection: A survey". In: *arXiv preprint arXiv:1901.03407* (2019).

[43] Saikiran Bulusu et al. "Anomalous Example Detection in Deep Learning: A Survey". In: *IEEE Access* 8 (2020), pp. 132330–132347. DOI: 10.1109/ACCESS.2020.3010274.

[44] Jiefeng Chen et al. "Robust Out-of-distribution Detection for Neural Networks". In: *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*. 2022.

[45] Mang Ye et al. "Deep learning for person re-identification: A survey and outlook". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[46] Bahram Lavi et al. "Survey on Reliable Deep Learning-Based Person Re-Identification Models: Are We There Yet?" In: *CoRR* abs/2005.00355 (2020).

[47] Khawar Islam. "Person search: New paradigm of person re-identification: A survey and outlook of recent works". In: *Image and Vision Computing* 101 (2020), p. 103970. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2020.103970.

[48] Qingming Leng, Mang Ye, and Qi Tian. "A Survey of Open-World Person Re-Identification". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.4 (2020), pp. 1092–1108. DOI: 10.1109/TCSVT.2019.2898940.

[49] Jose Miguel Huaman Cruz et al. "Benchmarking person re-identification approaches and training datasets for practical real-world implementations". In: *Under review* (2021).

[50] Hao Luo et al. "Bag of Tricks and a Strong Baseline for Deep Person Re-Identification". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 1487–1495. DOI: 10.1109/CVPRW.2019.00190.

[51] Ejaz Ahmed, Michael Jones, and Tim K. Marks. "An improved deep learning architecture for person re-identification". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3908–3916. DOI: 10.1109/CVPR.2015.7299016.

[52] William Brendel, Mohamed Amer, and Sinisa Todorovic. "Multiobject tracking as maximum weight independent set". In: *CVPR 2011*. 2011, pp. 1273–1280. DOI: 10.1109/CVPR.2011.5995395.

[53] Bing Wang et al. "Tracklet Association with Online Target-Specific Metric Learning". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1234–1241. DOI: 10.1109/CVPR.2014.161.

[54] Yang Zhang et al. "Long-Term Tracking With Deep Tracklet Association". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 6694–6706. DOI: 10.1109/TIP.2020.2993073.

[55] De Cheng et al. "Part-aware trajectories association across non-overlapping uncalibrated cameras". In: *Neurocomputing* 230 (2017), pp. 30–39. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2016.11.038.

[56] Minxian Li, Xiatian Zhu, and Shaogang Gong. "Unsupervised Tracklet Person Re-Identification". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.7 (2020), pp. 1770–1782. DOI: 10.1109/TPAMI.2019.2903058.

[57] Zhengxia Zou et al. "Object Detection in 20 Years: A Survey". In: *CoRR* abs/1905.05055 (2019).

[58] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[59] Arnold W. M. Smeulders et al. "Visual Tracking: An Experimental Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (2014), pp. 1442–1468. DOI: 10.1109/TPAMI.2013.230.

[60] Wei Li et al. "DeepReID: Deep Filter Pairing Neural Network for Person Re-identification". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 152–159. DOI: 10.1109/CVPR.2014.27.

[61] Christian Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 4278–4284.

[62] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[63] Liang Zheng et al. "Scalable Person Re-identification: A Benchmark". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1116–1124. DOI: 10.1109/ICCV.2015.133.

[64] Joris Guérin, Anne Magaly de Paula Canuto, and Luiz Marcos Garcia Goncalves. "Robust Detection of Objects under Periodic Motion with Gaussian Process Filtering". In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2020, pp. 685–692. DOI: 10.1109/ICMLA51294.2020.00113.