Universidad Politécnica de Madrid

Object recognition and tracking for surveillance applications using deep learning techniques

Doctorado en Tecnologías y Sistemas de Comunicaciones

Anastasios Dimou

2020

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

DEPARTAMENTO DE SEÑALES, SISTEMAS Y RADIOCOMUNICACIONES.

Object recognition and tracking for surveillance applications using deep learning techniques

Autor: Anastasios Dimou
Director: Dr. Federico Álvarez García, Dr.Petros Daras

El tribunal nombrado para juzgar la tesis arriba indicada el día de de , compuesto de los siguientes doctores:

Presidente: _____

Vocal: _____

Vocal: _____

Vocal: _____

Secretario: _____

Realizado el acto de lectura y defensa de la Tesis doctoral, a de de , acuerdan la calificación de:

Calificación: _____

EL PRESIDENTE EL SECRETARIO LOS VOCALES

Universidad Politécnica de Madrid

Object recognition and tracking for surveillance applications using deep learning techniques

Doctorado en Tecnologías y Sistemas de Comunicaciones

Anastasios Dimou

2020

Summary

Video analytics has become one of the hottest topics for computer vision researchers in the last years due to the dramatic evolution that Deep Neural Networks (DNNs) have introduced to the field. Despite of the progress achieved, advanced analytics have not still achieved a plug and play status due to a number of challenges that can be attributed to diverse video content in terms of quality, different camera parameters, non-optimal camera installation, or unstable video content.

In this thesis, two main aspects have been explored to make video analytics more robust and performing. On one hand, the extraction of better appearance-based features to improve the performance of fundamental image analysis tasks (e.g. object detection and recognition, segmentation), and, on the other hand, the regularization of neural networks using multiple information modalities to enhance accuracy in every step of the video analysis process.

Convolutional Neural Networks (CNNs) are the backbone of every image analysis task. Constantly deeper and more complex architectures have been explored to improve the feature extraction process. Residual Networks (ResNets) have shown that better feature representation can be also achieved by minimizing the amount of information that has to be modeled. Based on this intuition, this thesis explores novel ResNet-based architectures to optimize the discriminative power of the extracted features.

Regularization techniques are an essential tool to improve the training process of complex data analysis tools, as they reduce the solution space enhancing performance and minimizing training time. Such methods are explored in this work to extract a robust optical flow by combining appearance and semantic information. The enhanced flow contributes towards improving the performance of video stabilization, object detection and multi-object tracking.

Moreover, a number of techniques have been introduced to streamline the performance of basic image analysis tasks. For object detection, an adaptive region proposal method to speed up the process and the use of motion cues to improve performance has been proposed. A multi-modal re-identification method to improve multi-object tracking in complex and multi-camera environments has been also presented.

Finally, all the developed solutions have been integrated in real applications, including a surveillance investigation assistant for video archives and a crowd behavior analysis system.

Resumen

El análisis de video se ha convertido en uno de los temas más relevantes en el campo de la visión artificial en los últimos años, debido sobre todo a la gran mejora que la utilización de

redes neuronales profundas ha traído a este campo. Aún así, la analítica avanzada todavía no ha podido lograr una fácil integración en cualquier sistema existente debido al gran número de dificultades que presentan como son: la diversidad de contenido de video en términos de calidad, parámetros diferentes en las cámaras, una instalación no perfecta de las cámaras o tener un contenido de vídeo inestable.

En esta tesis, se han explorado dos aspectos principales para hacer que la analítica de video sea más sólida y eficaz. Por un lado, la extracción de mejores características basadas en la apariencia del video, para mejorar el rendimiento de las tareas básicas de análisis de imágenes (por ejemplo, detección y reconocimiento de objetos, segmentación) y, por otro lado, la regularización de las redes neuronales utilizando múltiples modalidades de información para mejorar la precisión en cada paso del proceso de análisis de video.

Las redes neuronales convolucionales (CNN) son la columna vertebral de cada tarea de análisis de imágenes. Se han explorado arquitecturas cada vez más profundas y complejas para mejorar el proceso de extracción de características. Las redes residuales (ResNets) han demostrado que también se puede lograr una mejor representación de características minimizando la cantidad de información que debe modelarse. Basándose en esta intuición, esta tesis explora arquitecturas novedosas basadas en ResNet para optimizar el poder para discriminar de las características extraídas.

Las técnicas de regularización son una herramienta esencial para mejorar el proceso de entrenamiento de herramientas complejas de análisis de datos, ya que reducen el espacio de la solución mejorando el rendimiento y minimizando el tiempo de entrenamiento. Estos métodos se exploran en este trabajo para extraer un flujo óptico robusto combinando apariencia visual e información semántica. El flujo óptico mejorado contribuye a mejorar el rendimiento de la estabilización de video, la detección de objetos y el seguimiento de múltiples objetos a la vez.

Asimismo, se han introducido varias técnicas para optimizar la realización de tareas básicas de análisis de imágenes. Para la detección de objetos, se ha propuesto un método de región de interés adaptativa para acelerar el proceso y el uso de señales de movimiento para mejorar el rendimiento. También se ha presentado un método de reidentificación multimodal para mejorar el seguimiento de múltiples objetos en entornos complejos y de múltiples cámaras. Finalmente, todas las soluciones desarrolladas se han integrado en aplicaciones reales, y se presentan en la tesis 2 ejemplos como son un sistema de videovigilancia con análisis de archivos de video, y un sistema de análisis de comportamiento de multitudes.

Keywords

Surveillance Camera Networks, Video Analytics, Object Detection, Object Tracking, Multi-object Tracking, Feature Extraction, Optical Flow Estimation, Motion Stabilization, Image segmentation, Image Classification, Event Detection, Multi-modal processing, Regularization, Data Fusion, Convolutional Neural Networks (CNNs), Residual Networks, Recurrent Neural Networks (RNNs), Linear Dynamic Systems, Crowd Behavior Analysis.

Palabras clave

Redes de cámaras de videovigilancia, análisis de video, detección de objetos, seguimiento de objetos, seguimiento de múltiples objetos, estimación de flujo óptico, estabilización del movimiento, segmentación de imágenes, clasificación de imágenes, detección de eventos, proceso multimodal, regularización, fusión de datos, redes convolucionales, redes residuales, redes neuronales recurrentes, sistemas lineales dinámicos, análisis de comportamiento de multitudes.

Agradecimientos

The journey towards successfully completing a PhD is long and demanding. It is important to have people that will accompany you in this journey and this section is all about them.

First of all, I would like to thank my two directors Federico Álvarez García and Petros Daras. Petros for believing in me and supporting me from the moment I joined VCL in 2011. He is one of the main motivators to even start the PhD and he provided me with the means and the space to complete it. Federico kindly accepted me in GATV and guided me during the PhD offering his expertise and advise in all stages of my work. Moreover, he supported me with all the difficulties arising from working remotely and not being a Spanish speaker. I really appreciate your efforts for me.

During the last 5 years, I had the privilege to collaborate with many colleagues in VCL with whom my publications were co-authored, namely Lazaros Lazaridis, Paschalina Medentzidou, Athanasios Psaltis, and Kosmas Dimitropoulos. I would like to separately thank Dimitrios Ataloglou and Konstantinos Karageorgos for the immense effort they provided to help me complete this PhD. Special thanks to Dimitrios Zarpalas, not only for his scientific contribution but as a friend for his continuous support and advice.

Finally, I would like to dedicate this work to my family; my parents, my wife and my children. I hope to make you all proud and to make up for all the time that I missed spending with you.

Thank you all.

Sincerely,

Anastasios Dimou

Index

Index of figures

Glossary

AP - Average Precision

ARMA - Auto-Regressive Moving Average

CAGR - Compound Annual Growth Rate

CCTV - Closed Circuit TV

CNN - Convolutional Neural Network

COCO - Common Objects in Context dataset

CUDA - Parallel computing platform and programming model developed by NVIDIA

DL - Deep Learning

DNN - Deep Neural Network

GPU - Graphics Processing Unit

IID - Independent and Identically Distributed

i-LIDS - Imagery Library for Intelligent Detection Systems

ILSVRC - Large Scale Visual Recognition Challenge

IOU - Intersection Over Union

IP - Internet Protocol

LDS - Linear Dynamical Systems

MAP - Mean Average Precision

MET - Metropolitan Police

MP - Megapixel

NMS - Non Maximum Suppression

PRID - Person Re-Identification Dataset

PTZ - Pan Tilt Zoom

RCNN - Region-Based Convolutional Neural Network

ResNet - Residual Networks

RNN - Recurrent Neural Networks

RoI - Region of Interest

RPN - Region Proposal Network

SoA - State of the Art

STN - Spatial Transformer Network

VGG - Visual Geometry Group

VMS - Video Management Systems

VOC - Visual Object Classes

ZF - Zeiler - Fergus

Chapter 1

Introduction

All around the world, organizations and agencies deploy video surveillance systems, driven by various factors such as increasing crime rate, security threats, terrorism acts, and extremism events. The rising number of such incidents have fueled the deployment of video surveillance systems in both public and private areas. In many countries, governments have implemented extensive surveillance systems for the protection of public places, aiming to answer those concerns. Sustained by this trend, the global video surveillance market is growing at a CAGR of 10.4% and is expected to reach $74.6 billion by 2025 [1]. Such an explosive growth is fueled by innovations in both hardware (camera technology, processing capabilities) and software (improved video analytics).

Modern surveillance systems utilize cameras that offer superior image quality that can reach up to 100 MP resolution while adapting to different lighting conditions. Their processing capabilities have also improved, offering the possibility to perform on-board image enhancement and analytics extractions. Some newer models are even integrating multiple GPU kernels to make this possible. However, the currently installed surveillance infrastructure (Figure 1.1) consists of a wide selection of cameras ranging from almost obsolete analogue cameras to multi-MegaPixel ones. Given that most of them are connected to Video Management Systems (VMS), the operators strive to take advantage of advanced technologies such as Artificial Intelligence, Big Data analytics, and cloud based services to enhance the current capabilities of the installed systems. The heterogeneity of the surveillance footage available poses, though, significant challenges for video analytics. In addition, mobile camera systems are becoming popular among security operators, posing even more challenges (e.g. video stabilization).

Figure 1.1: CCTV operations room

In this work, we are interested in addressing the fundamental challenges that are faced during the extraction of video analytics. More specifically, we are interested in improving the object detection capabilities of video analysis algorithms to address challenges posed by diverse video content in terms of quality and resolution. Moreover, the tracking of those objects in the scene and across surveillance cameras is investigated. Consistency in tracking is particularly important while extracting event analytics. Furthermore, the role of the optical flow in object detection and video stabilization is explored. Given that motion in video can be present due to the camera motion or objects' motion, the ability to distinct the perceived from the actual motion constitutes an important and challenging problem for video analysis applications.

In order to overcome these challenges, we investigate novel Deep Learning (DL) network architectures that improve the learning capacity of the network, achieving more accurate object detection and recognition. Moreover, an analysis of optical flow estimation networks is performed and semantic information is proposed to regularize them and improve their performance. The optical flow is utilized to improve the detection capability of objects and to stabilize videos from moving cameras. Finally, multi-object tracking is explored, using multiple information modalities, to achieve object re-identification after occlusions and across

cameras.

1.1 Motivation

The number of cameras installed by private and public bodies, for monitoring critical infrastructure, public places, office buildings, and private homes, is increasing day by day, exploding the volume of the footage produced daily. According to the industry analyst IHS Markit, in 2019 approx. 770 million professionally installed video surveillance cameras were active and operational globally, setting to become 1B by the end of 2021. The influx of surveillance footage from a growing number of cameras operating at constantly higher resolutions is posing extreme pressure both in terms of monitoring and storage. Especially for monitoring tasks that require intense and sustained attention, human operators tend to gradually lose their attention, a phenomenon also known as vigilance decrement. As time passes, the ability of the operator to detect a person, object or any other change in the scene is weakening. This is typically happening after 20 to 30 minutes of continuous work [2]. However, the human resources required to monitor the video feeds are simply not realistic, rendering the surveillance infrastructure useless.

Video analytics are destined to take the role of automatically monitoring the video feeds and alerting the operators for suspicious events, allowing them to focus on taking action if something goes amiss. While humans are bound by their physical and physiological limitations (visual capacity, tiredness, loss of concentration, etc.), computer vision solutions are limited only by the quality of algorithms and the processing infrastructure available. In view of those facts, sectors that have invested heavily in surveillance infrastructure are keen to exploit video analytics solutions for the automation of surveillance procedures. Such sectors include, among others, law enforcement agencies, municipalities and urban surveillance projects, mass transit monitoring and security, critical infrastructure protection, military and intelligence applications. The video analytics market is growing at an estimated Compound Annual Growth Rate (CAGR) of 21.5%, from $3.2 Billion in 2018 to $8.6 Billion by 2023 [3], even faster than the video surveillance market it belongs.

Video analytics have been marketed from the early 2000s, primarily, as alerting solutions. By triggering calls to action, they aimed to eliminate the need for active human video monitoring. Market offerings included the detection of an object or entity and recognizing its type, tracking of objects in the scene, the presence of motion in the scene, and car plate recognition.

Moreover, some more advanced functionalities included facial recognition (in a constrained environment), perimeter protection, people counting, abnormal event detection, and dynamic masking to block part of the video signal for privacy reasons. In practice, though, these video analytics encountered a lot of problems that hindered their wide acceptance from the customers. Despite the fact that they were installed and tuned by highly experienced personnel, their performance was sub-optimal due to their over-sensitivity to lighting conditions and noisy environments. Finding heuristically the correct thresholds to balance between low false alarms and high incident recall rate was not always possible. Therefore, despite their hype, they did not achieve the aim of removing human involvement in video surveillance.

In 2012, computer vision entered the DL era. Following a series of advances in both hardware and software, Deep Neural Networks (DNNs) were employed, dramatically improving many vision benchmark records. In terms of hardware the acceleration of computations required for DNNs using Graphical Processing Units (GPUs) reduced training times from weeks to days. In parallel, software implementations that used max-pooling to implement Convolutional Neural Networks (CNNs) were introduced. In October 2012, Krizhevsky et al.[4] won the large-scale ImageNet competition by a significant margin over shallow machine learning methods, anchoring a revolution that transformed the computer vision industry. Since then, our capabilities in image and video analysis tasks are continuously improving.

The video analytics industry took a quantum leap forward by leveraging DL. Given the availability of cluster/cloud computing infrastructure featuring GPUs, it became easier to leverage DNNs for analyzing images and videos. Modern video analytics solutions use DNNs to transform live or archived video into structured data with rich semantic metadata. Beyond the alerting functionality that was offered before, DL-driven solutions make it possible to uncover quantifiable data and trends from video metadata, offering deeper insight from previously underutilized video. Therefore, video system operators are enabled to review hours of video in minutes and rapidly identify people, objects and events of interest.

By leveraging powerful algorithms, video streams are analyzed to quickly identify and examine potential issues, anomalies, or particular events of interest. DL-driven solutions allow object extraction, recognition, classification, and indexing, for making video searchable, actionable and quantifiable. A non extensive list of features includes: detecting and tracking specific people/objects across multiple cameras, searching for faces matching a picture, performing behavior analysis, counting specific object types, number plate recognition, and extracting analytics for a specific person/object (e.g. gender, age, color, speed, pose, etc.). Moreover, DL is fueling a move towards a predictive approach aiming at identifying problems before

they happen through real time detection of known subjects or behavioral patterns.

The video surveillance sector offers numerous applications for visual analytics based on artificial intelligence. Machine learning and deep learning algorithms convert video surveillance systems into authentic visual intelligence ecosystems capable of recognising and analysing events and behaviour with an accuracy and rapidity that were impossible by the human operator alone. The applications are unlimited: detection of violent or abnormal behavior, detection of unruly crowd behaviour during events, prevention of accidents in public places (e.g. people falling onto the tracks in the metro) road safety assurance, remote area surveillance, etc..

Beyond security applications, video analytics can have a catalytic effect on a variety of different application fields. Business analytics in retailing regarding customer behavior constitute key information for marketing purposes. Automated retail stores without employees or check out processes are already being tested. Health monitoring is also an important field where video analytics can help, including baby monitoring systems, home rehabilitation and fall alerting for elderly people. It is, therefore, clear that the development of robust video analytics is of paramount importance, being able to radically transform a number of application sectors.

1.2 Problem description

Video analytics are destined to automate surveillance but in spite of the progress achieved, advanced analytics have not still achieved a plug and play status. They are facing a number of challenges that can be attributed to diverse video content in terms of quality, different camera parameters, non-optimal camera installation, and unstable video content.

Surveillance cameras are often poorly maintained (Figure 1.2), adding up to the challenge of addressing the pre-existing diversity [5]. A poor lens, dirty, with scratches or smudges can cause blurry image, removing or deforming object contours. Raindrops on it can also distort the image or cause light reflections. Direct light sources can cause flare or reflections covering the image and decreasing object clarity. Moreover, barrel distortion can deform objects making them unidentifiable to video analytics. Some examples of such footage are given in Figure 1.3.

As discussed in [6], most video analytics applications are following a pipeline of well established processing steps, depicted in Figure 1.4. The processing steps include image pre-processing,

Figure 1.2: CCTV cameras not properly maintained.

object recognition, object tracking, and activity recognition. The intermediate output results of each processing step are also shown. Analytics applications in specific domains may not employ all these steps, or may not apply them strictly in the order specified.

Image pre-processing is the process of preparing the frame for analysis. Given that the acquired data are diverse and they come from different sources, they need to be standardized and cleaned up before being fed to a neural network. Pre-processing is used to reduce the complexity and increase the accuracy of the applied algorithms. This step may include operations like cropping, de-warping, scaling, color correction, video stabilization or even super-resolution techniques [7] to enhance performance.

Object detection and recognition is one of the most studied problems in literature. DL algorithms are shown to surpass human level accuracy in predefined datasets, but in real conditions this not yet true. In cases where an object's appearance is heavily degraded, as in Figure 1.3a, there are still significant challenges that are connected with the capacity of the utilized DNN, the training procedure, and the regularization of the network. Moreover, the use of other modalities, such as motion, to improve our capabilities has not been adequately explored.

(a) **(b)**

Figure 1.3: Illustration of images where objects are not identifiable due to (a) dirt on the camera dome, and (b) bad lighting conditions.

Object tracking deals with creating a correspondence of an object along successive frames. The correspondence can be established using appearance and optical flow information. Challenges in tracking may refer to significant appearance changes due to pose and orientation changes or to very fast motion that makes it hard to establish a correspondence. Another significant challenge in tracking occurs due to occlusions. These occlusions can be due to (1) occlusion with background element of the scene (e.g. a person walking behind a tree and re-appearing), (2) inter-object occlusion (e.g. a person stepping out from the car, the person being in front of the car) (3) self occlusion where articulated objects can have parts of them occlude one another. Track re-identification to acquire it beyond an occlusion is a major issue for video analytics. The problem further scales up when multiple objects are tracked in a single scene.

Activity recognition is classifying the behavior of the objects that have been detected, and tracked in the context of the scene. At a high-level, activities can be described based on the properties of the object. Examples of high level activities include a loitering person, a fallen person, or a slow-moving vehicle. Additional part-based analysis may be done for increased levels of granularity, identifying activities, such as jumping, crouching, etc., motion patterns, such as gait. Events that include multiple actors can also be detected. The amount and diversity of activities that need to be detected is the most important challenge for robust activity/event detection in video analytics.

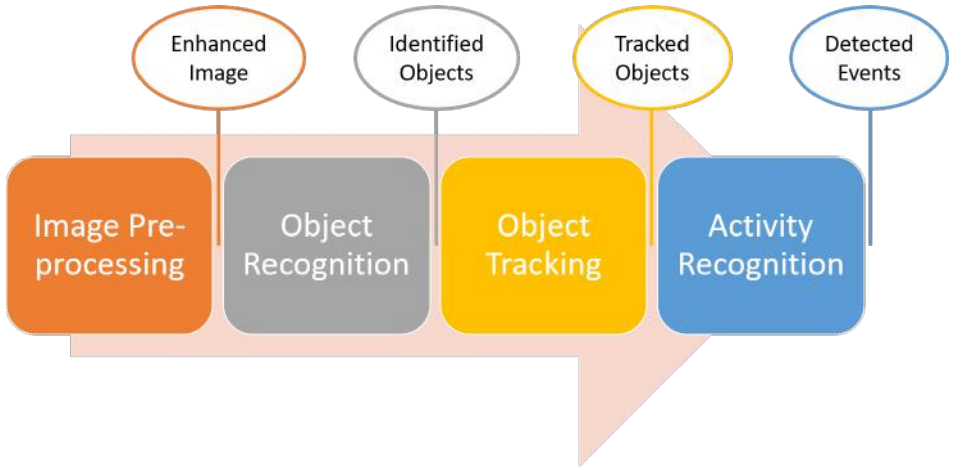


Figure 1.4: Video analytics processing pipeline.

1.3 Proposed approach

A synopsis of our approach towards improving the performance and the robustness of object recognition and tracking, as well as video analytics in general, is provided here. Starting from Figure 1.4 that depicts the basic steps of the video analytics pipeline and their respective output, we have identified key areas that require attention. Given a sequence of frames from a surveillance camera, we aspire to improve the existing video analytics capabilities by contributing to the following objectives:

1. Extract better appearance-based features to improve the performance of fundamental image analysis tasks.
2. Extract better optical flow information to improve motion-related tasks.
3. Improve object recognition accuracy and efficiency in challenging environments.
4. Enhance multi-object tracking in complex environments.
5. Enable video analytics for mobile camera devices producing unstable content.
6. Demonstrate real-world applications of the contributions.

1.3.1 Contributions towards the objectives

The contribution of this work towards each objective is illustrated as a processing block on the video analytics pipeline in Figure 1.5. In this section they are shortly presented, highlighting the publications that are supporting the validity of the proposed methodologies.

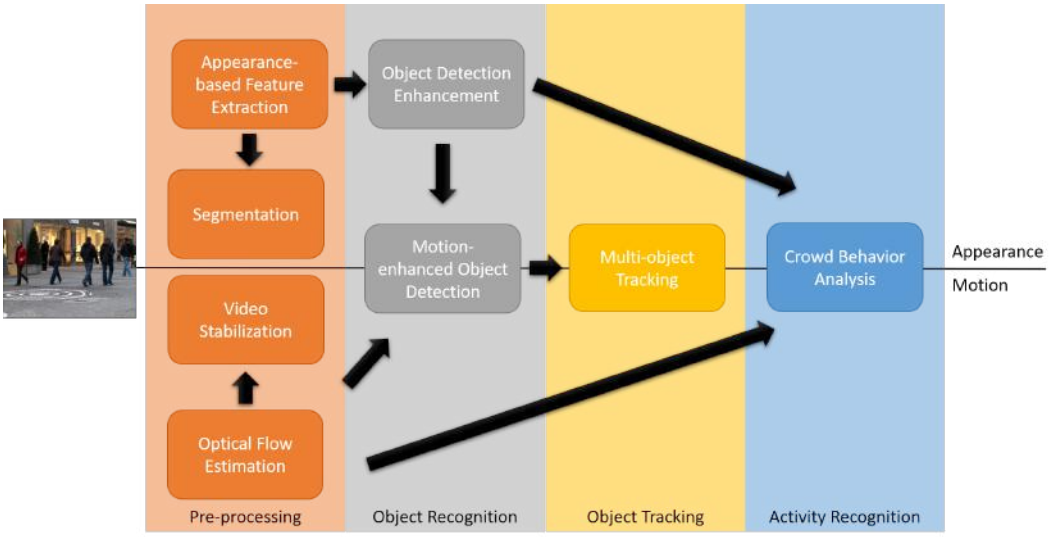


Figure 1.5: Overview of the approach and the contributions of this thesis.

The proposed pipeline is initiated with the pre-processing step that includes modules dealing with both the appearance and the motion. Chapter 3 is dealing with improving the appearance features extracted for image analysis task. A novel architecture based on ResNet and inspired by Linear Dynamic Systems is used to extract rich appearance features. The extracted features are shown to improve performance on image analysis tasks, including image classification and object recognition. This work was presented in [8].

Moreover, a CNN-based image segmentation architecture was developed using an iterative refinement scheme to produce a pixel-level object segmentation mask. The proposed method identifies both systematic and random segmentation faults dealing with them in a structured way. The appearance features extracted by the LDS ResNet proposed are also used. This work was presented in [9] significantly outperforming ResNet [10] in multiple vision tasks, showing complementarity with other improvement methods.

In Chapter 4 we are dealing with motion information. A novel optical flow estimation module is developed to estimate the flow between subsequent frames. A SoA optical flow estimator (FlowNet 2.0) is enhanced using regularization techniques to improve the training process. Semantic information extracted from segmentation masks and modified convolutional layers are utilized to guide the network to better performance. This work was presented in [11] outperforming the well-established FlowNet 2.0 [12] in multiple synthetic and realistic datasets.

The last contribution to the pre-processing step deals with videos from mobile devices that are highly unstable. A novel method that utilizes the optical flow in conjunction with the segmentation mask, to produce a stabilized version of the video is described in Chapter 7. Semantic information is used to efficiently filter out motion content that will disrupt the correct calculation of the intrinsic camera motion. This work was presented in [13] significantly improving the subjective quality of the output in challenging scenarios.

Moving to the object recognition step, new methods to improve both the efficiency and the performance of the task in Chapter 5. For this purpose, an RCNN-based network is used that comprises a feature extraction network, a region proposal mechanism and an object classifier. A novel methodology is proposed that predicts the size and position of previously detected objects, dynamically adjusting the region proposal parameters to enhance the procedure in terms of speed and accuracy. This work was presented in [14] demonstrating improvements in performance on challenging real-world surveillance videos including PTZ operations.

Moreover, a novel method that incorporates not only appearance but also motion in the object classification scheme is proposed to enhance the object recognition procedure. This neuroscience grounded method is built upon a SoA architecture that attempts to learn multiple objectives in a single training scheme. This work has been presented in [15] outperforming SoA methods in the object recognition task on multiple urban scene understanding datasets.

The object tracking step is analyzed in Chapter 6. A multi-object tracking method is presented to establish a correspondence of the objects along frames and define their trajectory. This is achieved combining multiple information modalities, including appearance features, positioning, velocity and social interaction with other objects in the scene. Object association is performed on a tracklet (i.e. short robust trajectories) basis, combining high performance and efficiency.

In the activity recognition step, the detected objects, their trajectory and the optical flow can be used to build diverse video analytics. In this work, A crowd behavior classification tool has been developed. Crowd density maps, extracted using CNN appearance features, are used

used as attention maps, in conjunction with the optical flow, to identify abnormal events in crowded areas. This work, presented in [16], SoA results were produced on existing datasets and a new synthetic dataset was produced and shared to advance research activities on the area.

1.4 Outline

The rest of this thesis is organized as described in the following. An analysis of relevant scientific literature is presented in Chapter 2. The main methodology and experimental evaluation carried out for each part are described in the following Chapters. Specifically, a novel CNN architecture to better model visual information is presented in Chapter 3. A novel method to estimate more accurately optical flow is presented in Chapter 4 towards improving the handling of both intrinsic and extrinsic motion of the video content. Moreover, novel and efficient methods for improving the performance of object detection by region proposal reasoning and incorporating motion flow information are analysed in Chapter 5. Chapter 6 presents a methodology for robust multi-object tracking and Chapter 7 deals with the stabilization of video content to improve the efficiency of video analytics. Real applications that use the proposed methodologies are presented in Chapter 8, while Chapter 9 concludes this thesis by providing a discussion on the impact and the limitations of the proposed framework, and pointing out research directions for future work and goals.

Chapter 2

Literature Overview

An overview of the literature related to the methods developed in this work is presented in this chapter. Established and state-of-art methods on convolutional neural networks, optical flow estimation, image segmentation, object detection, multi-object tracking, video stabilization and event detection in crowded scenes, are discussed.

2.1 Convolutional Neural Networks

CNNs are the "backbone" of image and video processing. They are used to encode the visual input into a feature representation that fits the task at hand in terms of accuracy and efficiency. The first CNN architecture, called LeNet, appeared on 1998 [17], but it wasn't until AlexNet [4] highlighted the potential of CNNs for image classification purposes, winning by a large margin the prestigious ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [18] on 2012. VGG [19] was the first network architecture that employed deeper networks progressively trained. This work initiated the development of two discrete neural network architectures, GoogLeNet [20] (a.k.a. Inception) and Deep Residual Networks (ResNet) [10]. Both architectures have evolved significantly since their appearance and they are the base for the current SoA architectures.

Since the appearance of ResNet, the concept of residual learning has been extensively utilized and extended in various ways. The current literature includes papers that improve perfor-

mance by easing the learning procedure, adding complexity to the network architecture or modifying each filters contribution.

In [21], authors proposed Wide Residual Networks (WRNs) to address the training difficulties and saturated performance of increasingly deeper models. WRNs were constructed by simultaneously decreasing the depth and increasing the width of residual networks, using a higher number of convolutional kernels and feature maps. These modifications resulted in much shallower networks that could generally achieve better performance than their deeper counterparts, while being easier to train.

In another attempt to alleviate training related inefficiencies in deep neural networks, such as vanishing gradients, diminishing feature reuse and long training times, the authors in [22] followed a different approach. Starting with very deep neural networks, they proposed to stochastically drop a percentage of the residual units in each training iteration and replace them with identity connections. This method, which extended the functionality of Dropout [23] to residual blocks, was able to reduce training time and continue improving the test error in residual networks featuring more than 1000 layers.

In ResNets, the input of each residual unit is forwarded to the next one through the identity shortcut and added to the unit's output. Thus, identity connections exist only between successive units and input information is merged and altered before it can be utilized by subsequent units. In *DenseNets* [24], the feature-maps of all preceding units are concatenated and used as inputs. Therefore, each unit output is used as input by all subsequent units. Concatenating feature maps learned by multiple different layers increases feature reuse throughout the network, which leads to improved training efficiency and lower classification error.

In [25], a deep dual-stream architecture was proposed, named "'ResNet in ResNet" (RiR), composed of generalized residual units that combine intersecting residual and transient paths. The generalized residual units retained the benefits of residual learning, while improving expressiveness and the ease to remove unnecessary information.

In order to boost the representational power of the network, [26] focused on inter-dependencies between the different channels of convolutional outputs and proposed the "Squeeze and Excitation" (SE) block, which adaptively re-calibrates channel-wise feature responses. Through feature re-calibration, the network uses global information and learns to selectively emphasize informative features and suppress less useful ones. Combined with residual connections, SE-ResNet units have set a new record in the ImageNet classification challenge.

2.2 Optical Flow Estimation

Optical flow estimation is a prerequisite step in a variety of computer vision problems, ranging from obvious ones, such as object tracking [27], action recognition [28], motion analysis [29] and video stabilization [30], to more sophisticated ones, such as monocular depth estimation [31], multi-frame super resolution [32] and 3D object reconstruction in immersive environments [33].

Regularization of optical flow estimation methods goes back to Horn and Schunck [34]. They formulated optical flow estimation as a global energy minimization task, which is intractable without regularization in the form of global motion consistency. To avoid the over-smoothing effect of total variation regularization, Shulman *et al.* [35] utilized a non-linear gradient weighting term. Nagel and Enkelmann [36] were the first ones to impose gradient weighting based on the local image structure with Werlberger *et al.* [37] combining the two in an anisotropic, image-driven Huber term. These methods make naive assumptions on the strength of each edge, given only its local structure and ignoring the global context of the scene. Such an approach could not suffice, since even rigidly moving objects can contain edges with arbitrary weights.

With the advent of deep learning, optical flow estimation methods using DNNs were proposed and the effort of the research community shifted to their optimization. The first work on optical flow estimation using CNNs was presented by Dosovitskiy *et al.* [38] who modeled it as a supervised learning problem. This model, FlowNet, was followed by a plethora of other works that sought to translate the powerful results of CNNs to optical flow. Subsequently, the proposed architectures were modified to better fit the nature of the problem, effectively providing an implicit regularization. The contributions included, among others, changes in learning rate scheduling, joint learning schemes and pyramidal architectures.

Tran *et al.* [39] implemented a 3D convolutional architecture, while Ranjan *et al.* [40] built a light network that is effective on areas with large motion using a spatial pyramid scheme. This multi-scale approach eventually emerged as the most popular choice in many general backbone networks for feature extraction. Sun *et al.* [41] also utilized it to construct a differentiable cost-volume, which topped most benchmarks when combined with intermediate wrapping operations. Yin *et al.* [42] proposed a hierarchical probabilistic formulation, trying to estimate not only the flow vectors themselves, but also their local distribution. Such an approach is shown to be computationally intractable on a global basis and without a pyramid scheme.

FlowNet2 from Ilg *et al.* [12] focused on the merits of module stacking and curriculum learning. The authors claimed that progressively increasing the difficulty of training samples plays a critical role on accuracy, therefore they proposed a stacked, progressively trained architecture with intermediate image warping operations, setting a new SoA. Hur *et al.* [43] claimed that stacked architectures benefit from reusing the same module for iterative residual refinement, instead of stacking independent modules. Hui *et al.* [44] tried to reduce the computational requirements of Ilg's method. Their approach included warping operations on the feature pyramid level, along with a custom local convolution operation and cost volume construction. Sun *et al.* [45] investigated the impact of training procedure even further. By carefully scheduling the learning rate, they significantly improved the original FlowNet's accuracy. Zhai *et al.* [46] made extensive use of dilated convolutions and residual blocks and demonstrated their detail-preserving properties.

Explicit regularization is more commonly used in unsupervised learning methods, since the common reconstruction loss function is an unstable training objective. This can be implemented in various ways, such as a consistency loss term [47, 48], utilizing flow predictions of classical methods for initialization [49], as well as an adversarial network [50]. The reconstruction error is naturally a bad means to achieve good results on occluded regions, since there is nothing to reconstruct. Liu *et al.* [51] enriched the training data with artificial occlusions, significantly improving occlusion performance. Of great interest in this area is the work of Yang *et al.* [52], where they proposed a Conditional Prior Network that regularizes the output based on the input. In essence, the network captures the possible motion space of a given single input image. Mun *et al.*[53] employs coupled spatio-temporal consistency checks in order to improve optical flow, depth and ego-motion.

Several attempts have been made to exploit the interdependence of semantics and motion in order to apply some semantically-driven regularization. Ha *et al.* [54] propose a semantically guided loss that improves deformation estimation. Sevilla *et al.* [55] split the computation into three parts, based on the semantic category of each object. Their formulation considers three general kinds of moving entities, based on the irregularity of their motion: 'planes', 'rigid' objects and 'stuff'. Such explicit approaches rely heavily on the granularity and accuracy of the segmentation [56]. Cheng *et al.* [57] tried to capture the interdependence between the tasks of semantic segmentation and optical flow estimation, jointly training a network for video object segmentation and optical flow, while employing late feature fusion. Wang *et al.* [58] used semantic masks for superpixel refinement and built a semantic-guided superpixel distance metric in order to improve sparse matching and flow estimation accuracy at object boundaries.

The correlation of optical flow and semantic masks has been explored in literature by jointly training a deep neural network to produce both of them. However, such approaches require significant structural modifications of the architecture and induce the additional burden of having to deal with multiple steps or training objectives.

2.3 Object Detection

Research on object detection and recognition initially focused on handcrafted features and shallow trainable architectures. However, in most cases, these models consisted of complex hierarchies of low-level features and high-level detectors, which burdened both the speed and the performance of the method, with their results far below human performance. The recent introduction of data-driven architectures has given a great boost in the field. Deep Neural Networks (DNNs) have shown considerable potential and displayed remarkable results to traditional vision problems with the great advantage of near real-time processing.

Over the past few years, a broad number of techniques have been proposed, targeting object detection from still images or videos, while combining and integrating different approaches. This section analyses the different methodologies available in the literature for object detection from still images and videos, focusing on DL techniques. DL methods can be roughly divided in (a) region-based and (b) regression-based ones, depending on the processing steps they employ. Moreover, the utilization of the flow as an auxiliary modality has been proposed.

2.3.1 Region-based methods

Current region-based methods perform detection by classifying different regions, sub-windows or patches extracted from the image. Their pipeline is to produce region proposals and subsequently classify each proposal into different object categories [59, 60, 61, 62, 63, 64]. The proposed approaches employ different methods to identify the candidate regions, aiming to find the correct balance between exhaustive search and a fixed number of region proposals. One of the first attempts to utilize CNNs in object detection was the Region-based CNN (R-CNN) [59] in which a number of class-agnostic candidate regions are proposed and fed to a CNN to extract a fixed-length feature descriptor for each region. Thereafter, a unique linear Support Vector Machine (SVM) for each class classifies these regions based on their extracted descriptors. In [60], a Spatial Pyramid Pooling (SPP) layer is introduced, in order to remove

the fixed object size constraint of the network. The latter computes a convolutional feature map from the entire image only once and then pools features in arbitrary regions to generate fixed-length representations for training the detectors.

Built upon R-CNN success, the Fast R-CNN [61] targets the inefficiency of having to pass each of the candidate regions individually through the CNN by forward passing the input image to the network once, generating its feature map and applying Region of Interest (RoI) pooling for each of the candidate regions to extract their feature representations. Based on the previously mentioned methods, Faster R-CNN [63] introduced a trainable mechanism for the purpose of proposing candidate regions called Regional Proposal Network (RPN). Given a number of uniformly generated anchors across the image, the RPN distinguishes them between foreground and background before passing the former to the classifier. Moreover, Mask R-CNN [62] extended the Faster R-CNN by adding an extra head for segmentation and replaced the ROI pooling mechanism with the RoI align method resulting in higher accuracy predictions. In [65], T.-Y. Lin et al. proposed Feature Pyramid Networks (FPN) on the basis of Faster R-CNN. The latter presented a top-down architecture with lateral connections for building high-level semantics at all scales. Later, a variety of improvements have been proposed, including R-FCN [64] and Light-head R-CNN [66].

In contrast to region-based detectors, such as Fast/Faster R-CNN [61, 63], that run a sub-network operations for each region proposed, the region-based detector in [64] is fully convolutional with almost all computations shared on the entire image. Thus, speed is improved as calculations for each image region are re-used. The latter introduces position-sensitive score maps to address a dilemma between translation-invariance in image classification and translation-variance in object detection. Additional classifiers were added in [67] aiming at progressively increasing the IoU's of the proposed regions with the ground truth objects resulting in improved predictions.

2.3.2 Regression-based methods

Contrary to the R-CNN family methods, where region proposition and region classification are done by discrete modules, in one-stage methods the regions are generated and classified in a single forward pass. Methods in this category try to map directly the image pixels to bounding box coordinates and class probabilities. In particular, Liu et al. [68] describe a method for detecting objects in images, using a single deep neural network. The approach, named Single shot multi-box detector (SSD), discretizes the output space of bounding boxes

into a set of default boxes over different aspect ratios and scales per feature map location. At inference, the network estimates the probability of each object category to be present in each box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. In [69], a CNN-based technique is proposed, which models the problem of object detection as an iterative search in a multi-scale grid-space of all possible bounding boxes.

The YOLO [70] and the SSD [68] algorithms are the most representative one-stage/regression object detection approaches. Later, R. Joseph has made a series of improvements on the basis of YOLO and has proposed its v2 and v3 editions [71], which further improve the detection accuracy while maintaining a very high detection speed. Moreover, an approach for introducing additional context into the SSD model is described in [72], where a state-of-the-art feature extractor (ResNet-101 [10]) is combined with the aforementioned detection framework [68]. The proposed SSD+ResNet-101 architecture is augmented with a set of deconvolution layers in order to introduce additional large-scale context in object detection.

Although this category of methods offers faster performance compared to the RPN based ones, they are limited in terms of prediction accuracy due to the high imbalance between positive and negative regions fed to the classifier (the positive and negative terms refer to the presence and the absence of ground truth object, respectively). Lin *et al.*[73] addresses the imbalance by having the ambiguous regions contribute more in the loss calculation, thus valuing the hard examples more than the easily classified ones. To this end, the authors introduced a novel loss function named 'Focal Loss' by reshaping the standard cross-entropy loss so that the detector will put more focus on hard misclassified examples during training. Focal Loss enables the one-stage detectors to achieve comparable accuracy of two-stage detectors while maintaining a very high detection speed.

2.3.3 Flow-based object detection

Optical flow based techniques have been applied to video object detection tasks over the years, as the incorporation of temporal information can improve the feature quality and recognition accuracy. The majority of them incorporate a flow obtained with an Optical Flow algorithm in the visual analysis loop for marking the detected object in the video frame. In [74], a DL framework, called T-CNN, which incorporates temporal and contextual information from tubelets/boxes obtained in videos is presented, that propagates detection results across

adjacent frames according to pre-computed optical flows. Zhu *et al.*[75] proposed a flow-guided feature aggregation, an accurate and end-to-end learning framework for video object detection, which leverages temporal coherence on feature-level. They later enhance the visual features by employing an optical flow network to estimate the motion between the nearby frames and the reference frame. Recently, a unified approach was introduced, which is based on the principle of multi-frame end-to-end learning of features and cross-frame motion. It belongs to the category of feature-level methods, and introduces a "Spatially-adaptive Partial Feature Updating" to fix inaccurate feature propagation caused by flawed optical flow.

2.4 Multi-Object Tracking

The most commonly used methods in multi-object tracking are Multiple Hypothesis Tracking (MHT) [76] and the Joint Probabilistic Data Association (JPDA) filters [77, 78], which delay making difficult decisions while there is high uncertainty over the object assignments. The combinatorial complexity of these approaches is exponential in the number of tracked objects making them impractical for real time applications. Object tracking has been successfully extended to scenarios with multiple targets [79, 80, 81, 82, 83]. Multiple target tracking does not mainly focus on appearance models, as it happens with single target tracking where sophisticated appearance models have been constructed to track a single target in different frames. Although appearance is an important cue, relying only on appearance can be problematic in multi-object tracking scenarios where the scene is highly crowded or when targets may share the same appearance. To this end, some works have been improving the appearance model [84, 85], while others have been combining the dynamics and interaction between targets with the target appearance [86, 87, 88, 89, 90, 91, 92].

Object tracking with deep learning techniques has attracted considerably less attention in the past, partly due to the lack of sufficient training data. Li et al. [93] incorporated a convolutional neural network (CNN) to visual tracking with multiple image cues as inputs. In [94] an ensemble of deep networks has been combined with an online boosting method. In [95], a single-target online learning tracker is proposed to alleviate blurring. Another line of research exploits auxiliary data to train offline a deep network, and then transfers knowledge to object tracking. Fan et al. [96] proposed learning a specific feature extractor with CNNs from an offline training set. In [97] a deep learning tracking method is proposed that uses stacked denoising autoencoder to learn the generic features from a large number of auxiliary images. On 2015, Wang et al. [98] employed a two-layer CNN to learn hierarchical features from auxiliary

data, which models complicated motion transformations and appearance variations. In [99], a deep learning architecture learns the most discriminative features via a CNN exploiting both the ground truth appearance information and the image observations obtained online.

2.5 Video Stabilization

Video stabilization techniques can be roughly categorized regarding their underlying motion model as 2D and 3D methods. 2D stabilization methods use a cascade of geometric transformations (such as homography or affine models) to represent the camera motion, and smooth these transformations to stabilize the video. The type of smoothing can have a dramatic effect on the qualitative evaluation of the result. One early method [100] used simple low-pass filtering, which requires very big temporal support to eliminate unwanted low frequency shaking (e.g. walking). Dealing with that, Chen *et al.* [101] applied polynomial curve fitting on top of Kalman-based filtering. Gleicher and F. Liu [102] broke camera trajectories into segments for individual smoothing, following principles of cinematography. Grundmann *et al.* [103] encapsulated this idea into an elegant L1-norm optimization, while S. Liu *et al.* [104] split the frame into multiple segments, each with its own path, and applied a joint stabilization method.

3D methods use the estimated camera position in space for stabilization and are, thus, heavily reliant on the effectiveness of structure from motion algorithms. Although they give superior results on complex scenes with parallax and depth changes, they are computationally heavier and less robust. An example of early work is from Beuhler *et al.* [105], who used a projective 3D reconstruction with an uncalibrated camera for video stabilization. F. Liu *et al.* [106] used 3D point correspondences to guide a novel and influential 'content-preserving' warping method, whose efficiency was later improved on planar regions by Zhou *et al.* [107]. S. Liu *et al.* [108] used a depth camera for robust stabilization.

In the middle ground between the two, 2.5D methods compensate for the lack of 3D information imposing additional constraints. F. Liu *et al.* [109] built on the observation that feature trajectories from a projective camera lie on a subspace and smoothed its basis eigenvectors. There is an extension of this method for stereoscopic videos as well [110]. Goldstein and Fattal [111] leverage the epipolar relations that exist among features of neighboring frames. Wang *et al.* [112] represented each trajectory as a Bezier curve and smoothed them with a spatio-temporal optimization. Though more robust than 3D methods, 2D ones demand re-

liable tracking to construct feature trajectories. S. Liu *et al.* [113, 30] try to alleviate the problem of acquiring long trajectories by smoothing the pixel profiles instead.

2.6 Crowd Event Detection

A popular approach for abnormal event detection, due to the lack of abnormal training data, is to first learn the normal patterns, and then anomalies are detected as events deviated from the normal patterns [114]. The majority of the work on anomaly detection relies on the extraction of local features from videos, that are then used to train a normality model. Trajectories have been a popular feature, detecting statistically significant deviations from the normal class to identify an anomaly [115]. However, tracking is impractical for detecting abnormal events in a crowded scene. Spatiotemporal anomalies of local low-level visual features, such as the histogram of oriented gradients [116], the histogram of oriented flows [117] and optical flow [118], by employing spatiotemporal video volumes (dense sampling or interest point selection) [119] have been also proposed. However, these approaches are bound to the quality and the completeness of the training set.

Deep learning methods have been also employed in anomaly detection. Unlike classic vision methods, optimal features are learned from the dataset. In [120], a 3D ConvNet was applied on classifying anomalies, whereas in [121] an end-to-end convolutional autoencoder was employed to detect anomalies in surveillance videos with good results. However, operations are performed only spatially, even though multiple frames are fed as input, because the 2D convolutions collapse temporal information [122]. On the other hand, Long Short Term Memory (LSTM) models are better suited for learning temporal patterns and predicting time series data. In [123], convolutional LSTMs have been proposed to learn the regular temporal sequences in videos.

In the presented work, a two-stream architecture is being proposed that employs crowd density heat-maps and optical flow respectively to detect abnormal events. Each modality is fed to a network with convolutional LSTM layers to model the spatiotemporal patterns of the input. The network is trained to detect *Panic* and *Fight* events. A synthetic dataset has been created using the GTA V engine to train the proposed network.

Chapter 3

Enhanced Feature Extraction for Image Analysis

As described in the proposed approach, presented in Section 1.3, the first step towards object recognition and tracking for surveillance systems is the processing of the video frames to extract visual appearance features. The aim of this step is to encode all the information of an image or video in a machine understandable way, preserving the completeness and the generality of the representation. In the Deep Learning era, this is performed by using CNNs as a backbone network. A brief history of CNNs is provided in Section 2.1.

This chapter presents the main contributions to the research community towards improving image feature extraction. In the introduction, the intuition of the proposed method is described. Subsequently, the proposed architecture is fully described and validated in diverse applications such as image classification, object detection and image segmentation.

3.1 Introduction

While increasingly deeper networks are proposed in the literature for feature extraction, their performance gets saturated and even degrades rapidly after a certain depth [10]. Despite the intuition that a deeper model should perform better than its shallower counterpart, adding

more layers may lead to a higher training error. The difficulty to train such a deeper network stems from the increasing complexity and the diminishing returns. ResNets introduced a new approach for Image Recognition with Deep Residual Learning. ResNets led to winning entries in the 2015 ImageNet [124] and MS COCO [125] competitions, in important tracks such as image classification, object detection, and semantic segmentation. The robustness of ResNets has since been proven in diverse visual recognition tasks and non-visual tasks including speech recognition systems [126]. ResNets consist of many stacked "Residual Blocks". Each block can be expressed in a general form:

$$y_l = h(x_l) + F(x_l, W_l), \tag{3.1}$$

$$x_{l+1} = f(y_l), \tag{3.2}$$

where x_l and x_{l+1} are input and output, respectively, of the l_{th} block, and F is a residual function. The central idea of ResNets is to learn the additive residual function F with respect to $h(x_l)$. The key choice in [10] is to use an identity mapping as h and a ReLU function [127] as f. The former is realized by attaching an identity skip connection, $h(x_l) = x_l$.

Instead of trying to fit each stack of layers directly to a desired underlying mapping, ResNet layers are trained to fit a residual mapping. The original mapping $F_{orig}(x_l)$ is recast into $F(x_l) + x_l$. It is argued that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. Hypothesizing that an identity mapping is the optimal one, it would be easier to push the residual to zero than to fit an identity mapping by a stack of non-linear layers [10]. ResNets that are over 100-layer deep have shown state-of-the-art accuracy for several challenging recognition tasks.

In this work, Residual Networks are re-visited focusing on the residual block as it proves to be the key for their success, while the depth of the network seems to have a supplementary effect. ResNets proved that residual mapping is easier to model than the original one [21]. Building on this insight, it is argued that methodologies for better modeling the residual information are expected to further improve the performance of the network.

An effective strategy to better model the residual information is presented by Inception-like multi branch architectures [20]. Inception models are based on a split-transform-merge strategy, where the input is tunneled into specialized branches that can model a different aspect of the residual information and are then merged again. These lower-dimensional embeddings create a strict subspace of the solution space of a single large layer operating on a high-dimensional embedding. However, Inception modules are approaching the representational power of a high-dimensional embedding with a considerably lower computational complexity, allowing the exploration of deeper networks.

The proposed method shares its principles between the two most important families of neural networks, namely ResNets and Inception networks. From ResNets, the research community has gained the insight that it is easier to try to model a residual of the initial information content rather than its whole. On the other hand, Inception networks have proved that modelling different aspects of the information content, using dedicated branches with different spatial support, can lead to improved performance. Starting from Inception v4 [128], skip connections were integrated to Inception-like models, confirming the compatibility of the two approaches.

A new approach to break down the information content is proposed here, aiming to improve data modeling by a residual block. It is argued that exploiting data statistics, a new representation of the residual data can be constructed to improve residual information modeling. This representation can offer a more generic description of the underlying texture that is easier to model. Towards this end, we consider a sequence of convolutional layers as a Markov process, in which the statistical representation of the process in the future, i.e., in the next layer, is completely determined by the present state, i.e. the output of the previous layer. Inspired by Linear Dynamical Systems (LDSs) and their ability to estimate the output of such systems through two stochastic processes, we initially design a novel module that simulates an LDS process and then propose various LDS block architectures aiming to better estimate the output of the residual block. A 3-path block is proposed, featuring: (1) a skip connection, (2) standard convolutional layers and (3) LDS modules, aiming at better modeling the residual information and easing the training process.

In order to understand the role of LDS-like modeling in a ResNet block, an exploration of the architecture design phase is presented in Section 3.2. The proposed ResNet block, inspired by LDSs, is utilized to better model the residual information. An extensive exploration of different architectures including the proposed module is performed. Different network topologies are developed and tested to compare performance with the original ResNet for multiple network depths. Moreover, the complementarity of the proposed methodology with other ResNet extensions is assessed. It is shown that such ResNet blocks achieve significantly better accuracy compared to the original ResNet block in the image classification problem. Experiments suggest that LDS-ResNet performs better than ResNets with a similar number of parameters. Moreover, it is shown that the performance boost is additive to other extensions of the original network such as pre-activation and bottleneck architectures, as well as stochastic training and Squeeze-Excitation. The new residual unit design presents improvements in the image classification and object detection tasks in public datasets such as CIFAR-10/100, ImageNet, VOC and MOT2017.

3.2 LDS-inspired ResNets

3.2.1 Linear Dynamical Systems

In the literature, LDSs have been widely used for the analysis of multi-dimensional data, such as time series and dynamic textures [129, 130, 131, 132]. An LDS is associated with a first order Auto-Regressive Moving Average (ARMA) process with white zero mean independent and identically distributed (IID) Gaussian input and for this reason LDSs are also known as linear Gaussian state-space models. The stochastic modeling of both signal's dynamics and appearance is encoded by two stochastic processes, in which dynamics are represented as a time-evolving hidden state process $x_t \in R^n$ and the observed data $y_t \in R^d$ as a linear function of the state vector:

$$x_t = Ax_{t-1} + Bv_t \tag{3.3}$$

$$y_t = Cx_t + w_t \tag{3.4}$$

where $A \in R^{n \times n}$ is the transition matrix of the hidden state, while $C \in R^{d \times n}$ is the mapping matrix of the hidden state to the output of the system. The quantities w_t and Bv_t are the measurement and process noise, respectively, with $w_t \sim N(0, R)$ and $Bv_t \sim N(0, Q)$.

In general, LDSs attempt to associate the output of the system with a linear function of a state variable, while in each time instance the state variable depends linearly on the state of the previous time instance. Many researchers have attempted to introduce a non-linear observation function f in (3.4), i.e., $y_t = f(x_t)$, maintaining in (3.3) the linear state transition [133], while others have used non-linear functions in both stochastic processes [134]. Besides modeling time-evolving data, LDSs have been effectively applied to problems involving the analysis of spatial information, such as still images, where the evolution of data is performed in the spatial domain, i.e., in consecutive pixels, instead of discrete time instances [135].

3.2.2 LDS module

Motivated by the aforementioned, we aim to create a novel LDS-inspired module that simulates the operation of an LDS system in order to improve the residual mapping of each block in a ResNet. Towards this end, instead of using a single identity mapping, we introduce a new branch with stacked LDS modules in order to calculate the estimated output for each residual block.

More specifically, when used within a CNN, the input to the proposed LDS module corresponds to a feature volume, which is a 3-D tensor of size $h \times w \times d_{in}$, where h, w are the spatial dimensions and d_{in} is the number of feature maps. The proposed LDS module operates over 3-D patches $X_{t-1} \in R^{n \times n \times d_{in}}$ along with the spatial dimensions of the input tensor (we used $n = 3$ in our experiments). More specifically, after applying the first-mode unfolding to X_{t-1} and obtaining a 2-D matrix $x_{t-1} \in R^{n \times n \cdot d_{in}}$, the corresponding output $y_t \in R^{d_{out}}$ is calculated as follows:

$$x_t = A x_{t-1} \tag{3.5}$$

$$y_t = f(W, x_t) \tag{3.6}$$

where $A \in R^{n \times n}$ is a square transition matrix, $x_t \in R^{n \times n \cdot d_{in}}$ is a transformation of the unfolded input patch and $f()$ is a mapping function, which uses x_t and a set of trainable parameters W to calculate the output y_t. In (3.5) and (3.6), t refers to the current processing step within a CNN structure, while $t-1$ to the previous one. All patch-wise outputs y_t are concatenated to a single 3-D tensor at the output of the LDS module.

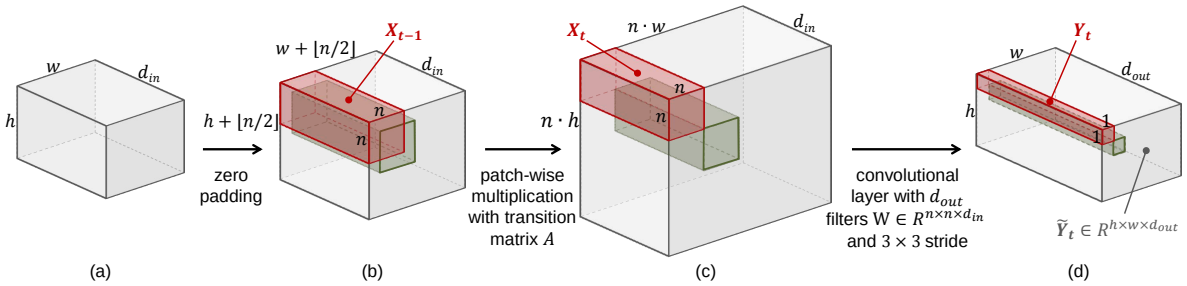


Figure 3.1: Graphical illustration of the proposed LDS module's operation. The 3-D volumes depicted in gray are: a) LDS module's input, b) zero-padded input, c) intermediate volume storing the multiplication products of each 3-D patch X_{t-1} with the transition matrix A and d) LDS module's output. The input to the LDS module is a 3-D tensor. In (b), the green patch is shifted by one position down and to the right compared to the red patch. In (c), their multiplication products with the transition matrix A are stored without overlapping. In (d), the outputs corresponding to each X_{t-1} are vectors with d_{out} elements.

The operation of the proposed LDS module is graphically presented in Figure 3.1. Calculating the LDS module's output $\widetilde{Y}_t$ involves three distinct steps. Firstly, in order to preserve the spatial dimensions between the input and the output tensors, the input tensor is zero padded along the spatial dimensions. When n is an odd scalar, the resulting tensor has a size of $(h + \lfloor n/2 \rfloor) \times (w + \lfloor n/2 \rfloor) \times d_{in}$. Then, we apply (3.5) individually to each 3-D patch X_{t-1} by

unfolding it to a 2-D matrix and multiplying by A. Multiplication products are folded back to 3-D subvolumes and stored successively without overlapping, resulting in an intermediate volume of $(n \cdot h) \times (n \cdot w) \times d_{in}$. Finally, this intermediate volume is convolved with a set of d_{out} filters $W \in R^{n \times n \times d_{in}}$, using a standard convolutional layer. Convolution is performed with a stride of n in each spatial dimension, in order to align the filters W with the regions corresponding to each of the 3-D patches X_{t-1} used in the previous step. In this way, the output of the LDS module $\widetilde{Y}_t$ is a $h \times w \times d_{out}$ tensor. The stride of the convolutional operation within the LDS module can be adjusted to multiples of n, which can prove useful in classification tasks, where the spatial dimensions of the input are usually reduced multiple times throughout the network. For example, a stride of $k \cdot n$ will reduce both the width and the height of the input tensor by k.

Figure 3.2: An LDS stack, composed of multiple LDS modules. The output of each LDS module can be directly used as input by the subsequent one.

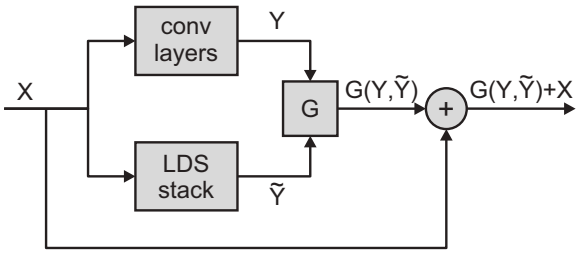


Figure 3.3: Abstract form of an LDS block, combining convolutional layers, LDS modules and an identity shortcut.

LDS modules can be stacked to form an LDS stack (Figure 3.2). In this case, the output of every LDS module is used as input by the subsequent module. A residual block can be modified by adding an LDS stack in a separate processing branch. Figure 3.3 shows the resulting LDS block in an abstract form, where the input X is processed by two parallel branches. Similarly to the original ResNet, the first branch consists of successive convolutional layers, while the second branch is either an LDS module or an LDS stack. The outputs Y and $\widetilde{Y}$ of the two branches are then combined by a function $G()$, before the addition of the input X, which is forwarded through an identity shortcut. The function $G()$ depends on the examined block type and is described in section 3.2.3.

3.2.2.1 Transition matrix calculation

We investigate two different approaches regarding the calculation of the transition matrix A. In the first one, its values are randomly initialized and subsequently optimized during CNN training through backpropagation. In this case, the values of A are trainable parameters, along with the filters of the convolutional layer. Aiming to preserve the spatial invariance property of CNNs, we always use the same matrix A when multiplying with values belonging to the same feature map of the LDS module's input tensor. On the other hand, we explored both the usage of a single matrix A for the whole input and d_{in} independent matrices, each dedicated to a specific channel of the input tensor.

In the second approach, inspired by [132], A is recalculated every time directly for each input patch X_{t-1} and is not optimized through backpropagation. Firstly, X_{t-1} is unfolded to produce a 2-D matrix. Starting from the first feature map, each $n \times n$ plane is placed to the right of the previous one, producing a 2D matrix $x_u \in R^{n \times n \cdot d_{in}}$ Then, A is calculated as:

$$A = X_2 X_1^T (X_1 X_1^T)^{-1} \tag{3.7}$$

where

$$X_1 = [x_u(1), x_u(2), ..., x_u(3d_1 - 1)] \tag{3.8}$$

$$X_2 = [x_u(2), x_u(3), ..., x_u(3d_1)] \tag{3.9}$$

and $x_u(i)$ is the i-th column of x_u.

3.2.2.2 LDS module implementation

The LDS module can be easily constructed utilizing a deep learning framework. It consists of three layers: a padding layer, a custom layer and a 2-D convolutional layer. The padding and the convolutional layers are included in most deep learning frameworks. For the latter, we used the CuDNN library [136] to speed up the training and inference processes. Calculation of the transition matrix A (when needed) and patch level multiplication (3.5) are executed by the custom layer. As this can become a computationally heavy operation for larger input sizes, the custom layer was implemented in CUDA to enable high level of parallelization. The code will be available upon the acceptance of this paper.

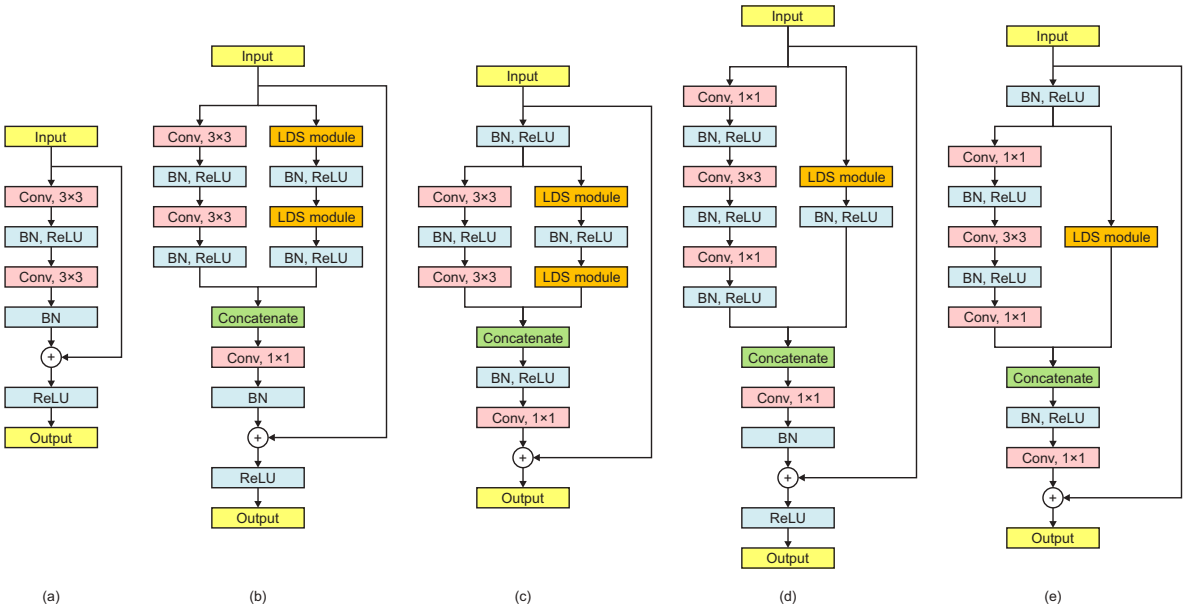


Figure 3.4: Block architectures: (a) baseline residual, (b) baseline LDS, (c) LDS with full pre-activation, (d) bottleneck LDS, (e) bottleneck LDS with full pre-activation

3.2.3 LDS blocks

In this section, the exploitation of the LDS module within residual blocks is explored in different setups. After describing the baseline residual block, different LDS block architectures are implemented and evaluated. An overview of tested block alternatives can be seen in Figure 3.4.

3.2.3.1 Baseline Residual block

The baseline residual block, depicted in Figure 3.4a, consists of two parallel branches. The first branch includes two 2-D convolutional layers with 3×3 filters, whereas the second one is a simple identity shortcut. Both convolutional layers use zero padding of 1 pixel and are followed by a batch normalization layer [137]. The outputs of the two branches are added and the second ReLU activation [127] is placed after the addition. When the number of feature

maps of the input is changed by the convolutional layers, either (A) the shortcut's output is padded with zeros to match the first branch's output, or (B) the identity shortcut is replaced by a projection shortcut, which consists of a 1×1 convolutional layer with the desired number of filters, followed by a batch normalization layer. When a convolutional layer reduces the spatial dimensions of the input by performing strided convolution, a spatial average pooling layer with the same stride is added in the identity shortcut branch or the same stride is employed by the 1×1 convolutional layer, in case of a projection shortcut.

3.2.3.2 Baseline LDS block

The baseline LDS block is presented in Figure 3.4b. The first branch of the baseline residual block is replaced by a two pathway structure. The first path is similar to that of the baseline residual block, including two convolutional layers, followed by batch normalization and ReLU activation layers. In the second path, the convolutional layers are substituted with LDS modules. Each LDS module has the internal structure described in section 3.2.2. The convolutional layers inside the LDS modules always use the same number of filters as the corresponding ones of the first path, but their stride is multiplied by 3. The outputs of the two pathways are concatenated along the third dimension, resulting in a volume with the same spatial dimensions, but double the amount of feature maps. An additional 1×1 convolutional layer is placed after the concatenation, which restores the number of feature maps to that of the input by blending the information received from the convolutional and the LDS paths. The identity shortcut is added after the concatenation and dimensionality reduction, following the same rules as in the residual case.

3.2.3.3 Pre-activated LDS block

In [138], authors suggested that moving the batch normalization and activation layers of a residual block before the convolutional layers can have a positive impact on the performance of a CNN. This pre-activation scheme has been employed in the LDS block to explore the complementarity of the two methods. Figure 3.4c shows how pre-activation was applied to the LDS block. In this case, the ReLU layer that followed the addition is moved before the 1×1 convolutional layer. This can prove beneficial in deeper networks, where such blocks are usually stacked, as information can be freely propagated through the shortcuts, without being altered by intermediate layers. Pre-activation is also applied in the case of projection shortcuts.

3.2.3.4 Bottleneck LDS block

Employing 3×3 filters on all convolutional layers can be computationally expensive for larger input sizes and deeper networks. The bottleneck architecture, originally proposed in [10], modifies the baseline residual block by stacking three convolutional layers instead of two. The first and third convolutional layers reduce and restore the number of features maps of the input volume, using 1×1 filters. The 3×3 convolution is performed on a smaller volume, which is usually equal to $1/4$ of the input. When the number of filters for the 3×3 convolutional layers is fixed, the two alternatives have roughly the same computational complexity, but the network consisting of bottleneck blocks will be deeper. Figure 3.4d shows the structure of the bottleneck LDS block. As there is only one 3×3 convolutional layer in the convolutional path, one corresponding LDS module is placed in the LDS path. Since the LDS module operates on patches along the spatial dimensions, placing more LDS modules to match the 1×1 convolutional layers would be unorthodox, as these would have to operate on 1×1 patches (both the patch and the transition matrix A would be a scalar value).

3.2.3.5 Bottleneck pre-activated LDS block

Our final block type combines the LDS module with both the pre-activation and the bottleneck schemes. The resulting bottleneck pre-activated LDS block is depicted in Figure 3.4e. Similar to the simple bottleneck case (Figure 3.4d), a single LDS module is used, while batch normalization and ReLU layers are moved before convolutional layers, as in Figure 3.4c.

3.3 Experimental Evaluation

In this section, the above described architectures are explored and validated in different datasets and applications that are related to object detection and tracking in video surveillance applications. Initially, the proposed networks are compared in the task of classification in CIFAR-10, CIFAR-100 and ImageNet. Subsequently, the proposed network is applied for object detection purposes on two different datasets, namely VOC and MOT Challenge 2017. Finally, LDS-ResNet is employed on the image segmentation task, proving its universal applicability in vision tasks.

Regarding the implementation, we utilized Torch 7 [139] and TensorFlow 1.4.0 [140] for the

experiments on classification and detection, respectively. The CuDNN library was used to speed up the model inference. Experiments were performed on a computer equipped with an NVIDIA GTX Titan X graphics card, CUDA v8.0.44 and CuDNN v5.1.5.

3.3.1 Performance evaluation on image classification

While image classification is not a core task for surveillance applications, it is often used as a pre-training step for object detection algorithms to provide a solid initialization of the backbone network. The experiments performed to assess the proposed LDS-ResNet architecture are presented here.

3.3.1.1 Image classification on CIFAR

The CIFAR-10 benchmark dataset [141] consists of 60,000 32×32 RGB images, classified into 10 classes. The dataset is split in 50,000 training and 10,000 test samples. Both training and test sets contain equal number of instances from each class. The CIFAR-100 dataset comprises the same images as CIFAR-10, except that they are divided into 100 finer classes.

We utilized the CIFAR datasets to study the effect of different block architectures and LDS module variants. We considered three different network depths for experiments using the baseline LDS block: 20, 44 and 110 layers. Additional experiments were performed with the pre-activated and bottleneck LDS blocks for the deeper network structures.

For all CIFAR experiments, we selected the overall CNN architecture that was used by [10], which is also presented in Table 3.1. A 2-D convolutional layer with 16 3×3 filters is placed first, followed by batch normalization and ReLU layers. The main network consists of 3 block groups with a total of $3k$ blocks. The number of filters used for convolutional operations is 16, 32 and 64, respectively. The second and the third block groups reduce the spatial dimensions by applying a stride of 2 in the first convolutional layers of their first blocks. Setting the value of k to 3, 7 and 18 leads to 20, 44 and 110-layer CNNs, respectively. For bottleneck architectures, $k = 18$ leads to a 164-layer deep CNN, as each bottleneck block has three convolutional layers instead of two. Finally, a 8×8 average pooling layer reduces the output size to a vector of size 64 and a fully connected layer produces the final class scores, which are 10 or 100 for CIFAR-10 and CIFAR 100 respectively.

Table 3.1: CNN architecture for CIFAR experiments

layers	# filters	stride	output size
input			$32 \times 32 \times 1$
convolution, 3×3	16		$33 \times 32 \times 16$
BN, Relu			
LDS block $\times$ k	16		$32 \times 32 \times 16$
LDS block $\times$ k	32	2	$16 \times 16 \times 32$
LDS block $\times$ k	64	2	$8 \times 8 \times 64$
average pooling, 8×8			$1 \times 1 \times 64$
fully connected			10 or 100

For non-bottleneck blocks, when the output of a block has different dimensions compared to its input, the identity shortcut is replaced by a pooling and/or padding operation (option A). For bottleneck blocks, a projection shortcut (option B) is used instead. When pre-activated blocks are used, additional batch normalization and ReLU layers are placed after the third block group. In this case, in the very first block of the CNN, the batch normalization and ReLU layers placed before the two paths in Figure 3.4c and 3.4e are omitted.

To enable fair comparison, we used the same training parameters as in [10] for all CIFAR experiments. Thus, training lasted for a total of 164 epochs, which is approximately 64k iterations with a batch size of 128 images. Initial learning rate was set to 0.1 and divided by a factor of 10 twice, at the beginning of epochs 82 and 122. CNNs were trained using Softmax loss and Stochastic Gradient Descent (SGD), with a momentum of 0.9 and a weight decay of 0.0001 was used for all trainable parameters. All images were normalized by subtracting the mean and dividing by the standard deviation computed over the entire training set. For data augmentation, training images were zero padded by 4 pixels along each direction and a random 32×32 patch was selected at each iteration. Additionally, we used random horizontal flipping with a 0.5 probability.

The following notations are used in Tables 3.2 to 3.6: *residual* refers to simple residual blocks, *LDS-1a* to LDS blocks where a single trainable transition matrix A is used for all feature maps, *LDS-1b* to LDS blocks with different trainable A matrices for each feature map and *LDS-2* to LDS blocks were A is calculated using (3.7). CNNs with pre-activated blocks are denoted with a *-pre* suffix after their depth.

Table 3.2: CIFAR-10 results (baseline and pre-activated blocks)

depth	residual	LDS-1a	LDS-1b	LDS-2
20	8.75 [10]	6.82	**6.75**	7.41
44	7.17 [10]	**6.24**	6.45	7.03
110	6.61 [10]	**5.74**	7.11	6.10
110-pre	6.37 [138]	5.48	**5.36**	5.73

Table 3.3: CIFAR-100 results (baseline and pre-activated blocks)

depth	residual	LDS-1a	LDS-1b	LDS-2
20	31.92	29.46	**29.30**	29.99
44	29.49	27.88	**27.36**	28.13
110	28.62	26.18	**26.15**	26.66
110-pre	26.92	25.47	**24.79**	25.70

3.3.1.2 Baseline LDS results

Table 3.2 presents the error rate in the CIFAR-10 test set for non-bottleneck ResNet and LDS-Resnet architectures. We observe that in most cases, CNN based on LDS blocks exhibit a significantly lower error rate compared to their residual counterparts. Table 3.3 shows the experimental results in the CIFAR-100 test set. In these experiments, all LDS blocks performed significantly better than the residual ones. Error rates with residual blocks were obtained from our experiments using the same experimental setup, as there were no relevant results for non-bottleneck architectures in [10] and [138].

The introduction of the additional LDS path increases the width and doubles the number of trainable parameters in LDS/ResNets. To investigate the possible impact on performance, we performed two sets of ablation experiments, following different approaches to construct alternative non/LDS models with the same depth and number of paramaters as LDS/ResNets. In the first one, the original ResNets were widened by using more filters in each convolutional layer. In particular, to double the overall number of trainable parameters, the number of filters in each convolutional layer was multiplied by $\sqrt{2}$. In the second approach, the LDS modules in Figure 3.4b were substituted with standard convolutional layers, resulting in 3-branch inception-like residual blocks, composed of two identical convolutional branches (which were concatenated in the same way) and the shortcut connection.

Table 3.4: Comparison between LDS-ResNet and alternative ResNet architectures with similar number of parameters (CIFAR-10)

	ResNet [10]		widened ResNet		3-branch ResNet		LDS-ResNet	
depth	params	error	params	error	params	error	params	error
20	0.27 M	8.75	0.55 M	6.94	0.57 M	6.85	0.57 M	**6.82**
44	0.66 M	7.17	1.36 M	7.58	0.39 M	6.63	1.39 M	**6.24**
110	1.73 M	6.61	3.56 M	6.75	3.65 M	6.35	3.65 M	**5.74**

A performance comparison is presented in Table 3.4. It can be seen that in deeper networks (44 and 110 layers), which are the ones that also achieve the lowest error rates, there are no performance benefits when widening the original ResNets. On the other hand, LDS-ResNets outperforms both alternative architectures by an increasing margin as the depth of the network increases. Thus, the performance benefits can be mostly attributed to the proposed LDS module, rather than the increase of the network's width and parameters.

3.3.1.3 Pre-activated LDS results

Results with pre-activated blocks are presented in Tables 3.2 and 3.3 for the 110-layer CNNs. In both datasets, the pre-activated block exhibits better performance than the baseline blocks, both for residual and LDS blocks. Overall, using non-bottleneck blocks, the best performing CNN in CIFAR-10 has an error rate of 5.36 (20% lower compared to its residual equivalent) and is 110-layer deep, composed of pre-activated LDS blocks with different transition matrices for each feature map (LDS-1b). Similarly, in CIFAR-100, the best performing CNN reduces the error rate from 26.92% to 24.79% using the same type of LDS blocks.

3.3.1.4 Bottleneck LDS results

Tables 3.5 and 3.6 depict the error rates using bottleneck blocks, in CIFAR-10 and CIFAR-100 test sets, respectively. It can be seen that all LDS block variants outperform residual blocks in both datasets, despite the fact that, due to GPU memory limitations, CNNs with LDS blocks were trained with a lower batch size of 64 images. Furthermore, the error rate is significantly lower compared to non-bottleneck blocks for all block variants. On the other hand, compared to non-bottleneck blocks, pre-activation has a more limited impact on performance, especially

Table 3.5: CIFAR-10 results (bottleneck blocks)

depth	residual [138]	LDS-1a	LDS-1b	LDS-2
164	5.93	4.57	**4.27**	4.40
164-pre	5.46	4.49	**4.39**	4.61

Table 3.6: CIFAR-100 results (bottleneck blocks)

depth	residual [138]	LDS-1a	LDS-1b	LDS-2
164	25.16	22.38	21.84	**21.24**
164-pre	24.33	21.96	21.53	**20.97**

when bottleneck LDS blocks are used. The best performing CNNs achieved an error rate or 4.27% in CIFAR-10, using LDS-1b bottleneck blocks and 20.97% in CIFAR-100, using pre-activated LDS-2 bottleneck blocks.

3.3.1.5 Combination and comparison with other ResNet extensions

A key advantage of the proposed LDS module is that, besides improving ResNet performance, it can be used complementary to other ResNet extensions and related methodologies. Results reported in subsections 3.3.1.3 and 3.3.1.4 highlight the complementarity of the LDS module with pre-activated and bottleneck CNN architectures. We build upon baseline LDS-ResNets and assess them in combination with the Stochastic Depth (SE) [22] and Squeeze-Excitation [26] methods. We also compare the performance of models combined with other ResNet extensions, selecting models with similar number of parameters when possible.

Table 3.7 presents the experimental results. For non-LDS models, results are reported as they appear in the respective publications, except for the SE method, where they correspond to our implementation of the method (optimized for the *compression rate* hyperparameter), as no results for the CIFAR datasets were available in the paper. Whenever stochastic depth was applied, we followed the training schedule used in [22].

As it can be seen, the addition of stochastic depth and SE alone significantly improves over both the respective non/LDS methods and the baseline LDS results. Nonetheless, stochastic depth has a greater impact than SE, achieving a test error of 4.07 in CIFAR-10, which already compares favorably to the rest of non-LDS ResNet extentions. Combining both stochastic

depth and SE with LDS further reduces the error in both datasets. While not yet achieving best result in CIFAR-100, it is important to note that the "LDS + stochastic + SE" model has significantly less trainable parameters (3.9 M) than DenseNet (7.0 M). Finally, we trained two wider models with approximately the same number of parameters as DenseNet, which reduced the CIFAR/100 test error to 19.56, thus, outperforming all entries of Table 3.6.

3.3.1.6 Training time

Training duration depends on the depth of the network and the usage of bottleneck blocks. For CNNs with baseline LDS blocks, training lasted on average for 2.4, 5.6 and 14.2 hours, for the 20, 44 and 110-layer models respectively. The 164-layer models with bottleneck LDS blocks, required 33.1 hours to complete the training process. As pre-activated blocks are composed of the same elements as regular ones and only alter the sequence of operations within each block, their introduction has only a negligible effect on training time.

3.3.1.7 Image classification on ImageNet

While experimentation with the CIFAR dataset provided the means to explore different architectures and to evaluate their performance, its very small sized images (32×32 pixels) prohibited from utilizing the trained networks in real-world surveillance applications, where higher resolution inputs are common. To that end, we utilized the ImageNet dataset [124], both to evaluate the classification capabilities of our method on higher resolution images and to use the trained models as a basis for our detection experiments.

The ImageNet 2012 classification dataset consists of 1.28 million training and 50k validation images of variable size, classified into 1000 classes. Each class contains 732-1300 training and exactly 50 validation images. The average image resolution is 469×387 pixels. Since training with such a large dataset can be time consuming, we extracted a subset of the ImageNet dataset by randomly selecting 500 classes and half of the training images for each selected class. All validation images from the 500 selected classes were used for testing. Thus, the selected subset has a total of 320k training and 25k test images. We performed experiments on the same subset using residual and LDS blocks, allowing fair comparison between them.

We chose the overall structure of ResNet-50 with bottleneck residual blocks as the baseline architecture for ImageNet experimentation. Then, all residual blocks were replaced by bot-

Table 3.7: Test error comparison in CIFAR-10/100

method	configuration	params	CIFAR-10	CIFAR-100
ResNet [10]	110-layer	1.7 M	6.61	28.62
Stochastic Depth [22]	110-layer, *drop rate* = 0.5 (linear decay)	1.7 M	5.25	24.98
Squeeze-Excitation [26]	110-layer, *compression rate* = 1 (our implementation)	1.9 M	6.10	26.08
DenseNet [24]	100-layer, *growth rate* = 12	7.0 M	4.10	20.20
Wide ResNet [21]	40-layer, 4× wide	8.9 M	4.53	21.18
ResNet In ResNet [25]	18-layer + wide RiR	10.3 M	5.01	22.90
LDS	110-layer, baseline LDS blocks	3.7 M	5.74	26.15
LDS + stochastic	110-layer, *drop rate* = 0.5 (linear decay)	3.7 M	4.07	21.44
LDS + SE	110-layer, *compression rate* = 1	3.9 M	5.56	25.13
LDS + stochastic + SE	110-layer, *drop rate* = 0.5 (linear decay), *compression rate* = 1	3.9 M	3.94	20.77
LDS + stochastic + SE (wider)	same as above but with 21 channels	6.7 M	3.90	20.03
LDS + stochastic + SE (wider)	same as above but with 22 channels	7.3 M	**3.87**	**19.56**

Table 3.8: CNN architecture for ImageNet experiments

layers	# filters	stride	output size
input			$224 \times 224 \times 1$
convolution, 7×7	64	2	$112 \times 112 \times 64$
BN, Relu			
max pooling, 3×3		2	$56 \times 56 \times 64$
bottleneck LDS block $\times 3$	256	2	$28 \times 28 \times 256$
bottleneck LDS block $\times 4$	512	2	$14 \times 14 \times 512$
bottleneck LDS block $\times 6$	1024	2	$7 \times 7 \times 1024$
bottleneck LDS block $\times 3$	2048		$7 \times 7 \times 2048$
average pooling, 7×7			$1 \times 1 \times 2048$
fully connected			500

tleneck LDS blocks (Figure 3.4d) with different transition matrices A for every feature map (variant LDS-1b), which were the ones that achieved the best results in CIFAR-10. The resulting architecture is presented in Table 3.8. The main network consists of four block groups, with different number of blocks and feature maps. Strides are always applied to the last block of the group. In these blocks, a max pooling layer with 1×1 filters and stride 2 is added replacing the identity shortcut. When the number of feature maps changes between blocks, projection shortcuts are used.

Preprocessing of training images involved random horizontal flipping with 0.5 probability, scaling with the smaller edge to equal a random size taken from a uniform distribution in the $[256, 512]$ range and finally, cropping a random 224×224 patch. Test images were scaled with their smaller edge to be exactly 256 pixels and a single 224×224 patch was later cropped from the center of the scaled image. The mean RGB value was subtracted from both training and test images.

Aiming to reduce the required training time, rather than maximizing the classification performance of the network, we limited the training procedure to 50 epochs, with the initial learning rate of 0.1 divided by 10 twice at the beginning of epochs 21 and 36. Due to the larger input size and the increased number of intermediate feature maps, a lower batch size of 32 was used to fit the LDS models in GPU memory. As in CIFAR experiments, all CNNs were trained using Softmax loss and SGD with 0.9 momentum and 0.0001 weight decay. Training on ImageNet, which was the most computationally expensive task of our experiments, lasted for almost 17 days, using a dual GPU setup with an additional NVIDIA Tesla K40c.

Table 3.9: ImageNet results

	residual	LDS-1b
top-1 error	32.29	**27.38**
top-5 error	11.97	**9.06**

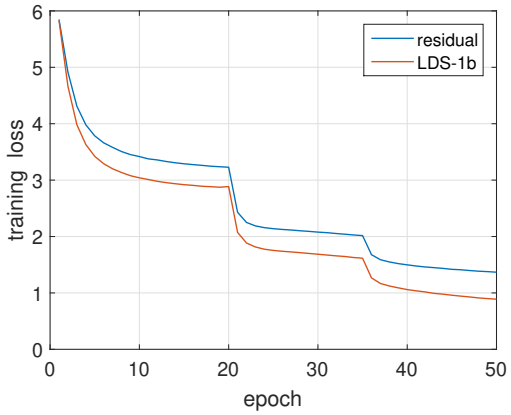


Figure 3.5: Convergence comparison in ImageNet classification using residual and LDS-1b bottleneck blocks.

Table 3.9 shows the top-1 and top-5 classification error in the selected test set. For the top-1 error, an image is considered to be classified correctly only when the true label matches the top scoring output class, whereas for the top-5 error, the true label should be among the five classes with the higher scores. LDS blocks reduce both top-1 and top-5 errors by 15% and 24% respectively. Figure 3.5 presents the convergence of the networks during training. It can be seen that the CNN containing LDS blocks converges faster and achieves a lower training loss.

3.3.2 Performance evaluation on object detection

Object detection is a key module for surveillance purposes as it allows us to extract semantic information from the observed scene. Given that LDS-ResNets can improve the performance of ResNet-based methods in any domain, when used as a backbone they are evaluated for generic object detection and person detection in particular.

Faster-RCNN [63] was selected to evaluate the performance improvement when the base network is replaced with LDS-ResNet. Faster-RCNN is a widely used architecture for object detection, using either VGG or ResNet as a base network for region proposal generation and labelling the detected objects. The base network receives the input image and outputs a set of feature maps. Based on this intermediate representation, a class-agnostic Region Proposal Network (RPN) suggests possible object locations. Finally, the region proposals with the higher "objectness" scores are forwarded to the classifier, which assigns a class label to each one of them and refines the bounding box. Its modular architecture makes it ideal for testing new neural network models.

The LDS-ResNets, pre-trained with the ImageNet dataset (as explained in section 3.3.1.7), were utilized as the base network of Faster-RCNN, including all layers of Table 3.8 up to the third block group. It has been shown that using a pre-trained network as a starting point improves performance and reduces training time. The Region Proposal Network (RPN) was placed between the third and the fourth block groups, leaving the final three blocks along with the average pooling and fully connected layers to be part of the classifier.

We utilized the PASCAL VOC [142] and the MOT2017Det [143] datasets to evaluate the proposed architecture. The PASCAL VOC detection dataset comprises variable sized RGB images with multiple ground truth bounding boxes of objects over 20 classes. As is common practice, we used the VOC2007 and the VOC2012 training and validation sets for training purposes, with a total of 16551 images containing 40058 objects. For evaluation, we used the VOC2007 test set with 4952 images and 12032 objects. The MOT2017Det training set comprises 7 videos with a total of 5316 frames, captured by static or moving cameras in various environmental conditions. Each frame contains on average 21.1 pedestrians, leading to over 112k ground truth bounding boxes in total. We split this set equally into two subsets, taking the first half of each video for training and the second half for testing.

Training was performed in an end-to-end manner by combining classification and box regression losses from the RPN and the classifier modules. Non Maximal Suppression (NMS) with an Intersection over Union (IoU) of 0.7 was utilized to reduce the number of highly overlapping region proposals at the RPN output. Training was performed using a single image as input to Faster-RCNN and a sampled subset of 128 region proposals as input to the classifier module, containing equal amount of objects and background cases. The 7×7 convolutional layer at the beginning of the network and the first block group were kept fixed during training. Also, all batch normalization layers were set to inference mode. Our Faster-RCNN implementation uses TensorFlow and is based on [144].

Table 3.10: Faster-RCNN Detection Results (mAP) using ResNet and LDS-ResNet base networks

dataset	residual	LDS-1b
PASCAL VOC	67.87	**73.34**
MOT2017Det	68.31	**69.51**

All images were resized with their shortest side to equal 600 pixels. Training datasets were augmented with horizontally flipped images. Using a single image as input, training lasted for 110k and 70k iterations for the PASCAL VOC and the MOT2017Det datasets respectively. Initial learning rate of 0.001 was reduced by a factor of 10 after 80k or 50k iterations. Trainable parameters were optimized using SGD with 0.9 momentum and a weigh decay of 0.0001. Training with the PASCAL VOC dataset required 74 hours, while the MOT2017Det dataset required 47 hours.

During testing, the top scoring 300 region proposals (after NMS) were forwarded to the classifier. Using this configuration, the proposed architecture operated at a rate of 0.8 images per second in both datasets. Following the official PASCAL VOC evaluation rules, an IoU higher than 0.5 between the detection and the corresponding ground truth bounding boxes was required for a detection to be counted as a True Positive. Additionally, each ground truth box could be associated at most once with a detection box, counting multiple detection of the same object as False Positives and any remaining non-associated ground truth boxes as False Negatives. Evaluation metrics in the MOT2017Det dataset were obtained according to challenge rules using the provided evaluation scripts.

Detection performance was quantified by Average Precision (AP). For each examined class, AP was obtained from the Precision-Recall curve by averaging the computed precision at 0.1 recall intervals. Table 3.10 presents the mean Average Precision (mAP) in both datasets, using Faster-RCNN with residual and LDS-1b blocks. Figure 3.7 shows the AP for each individual class in the PASCAL VOC dataset. It can be seen that LDS blocks outperform residual ones in every single class, leading to an overall performance boost of 5.51% in terms of mAP. Figure 3.6 compares the Precision-Recall curves for the pedestrian class in the MOT2017Det dataset, where LDS blocks also lead to superior performance. Detection examples in the PASCAL VOC and MOT2017Det datasets are presented in Figure 3.8 and 3.9 respectively. The results presented in this subsection are indicative of the expected performance benefit in other methods that employ Faster-RCNN, if the proposed LDS-ResNet is utilized as the base network.

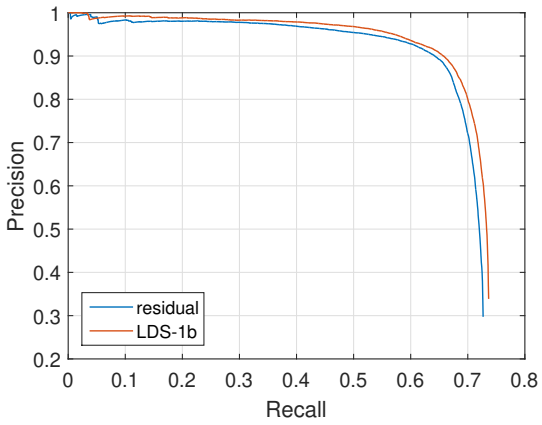


Figure 3.6: Precision-Recall curves in the MOT2017Det test set.

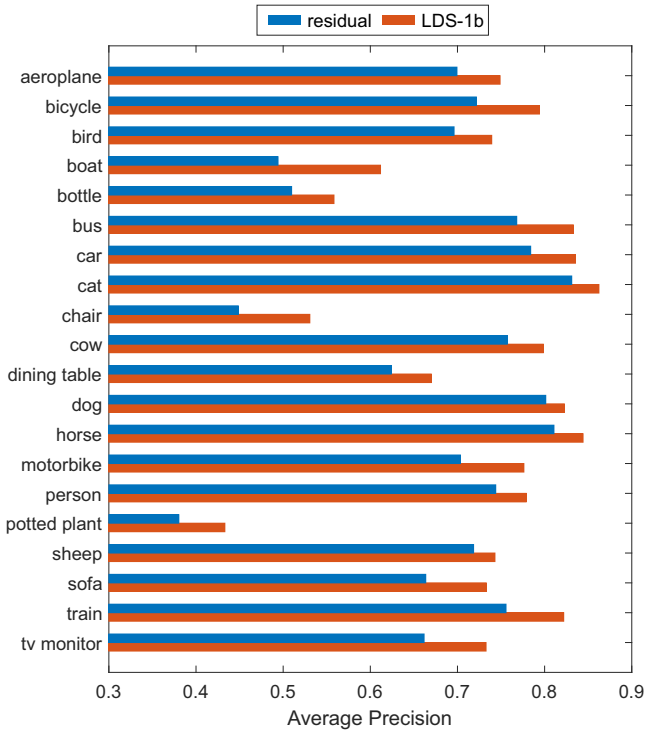


Figure 3.7: Average Precision per class in the VOC2007 test set

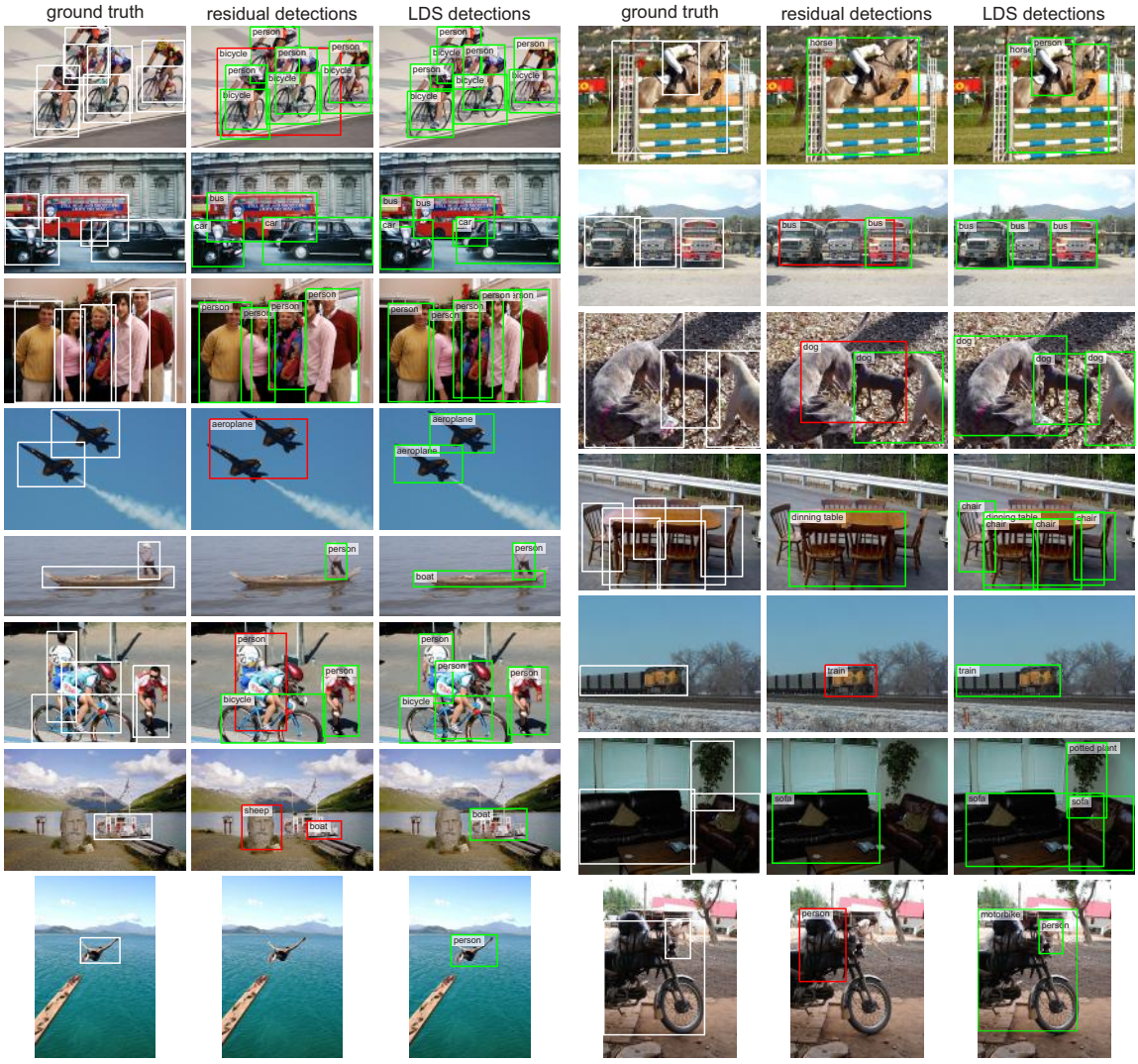


Figure 3.8: Detection examples in the PASCAL VOC dataset. Ground truth boxes are depicted in the left images, detections using Faster RCNN with residual blocks in the middle and detections using LDS blocks in the right ones. Detections counted as True Positives are presented with green bounding boxes, whereas False positives (either due to wrong class label or low IoU value with the ground truth) are presented with red. To obtain these detections, score thresholds have been optimized independently for each ground truth class and block type by maximizing the sum of precision and recall.

Figure 3.9: Detection examples in the MOT2017Det dataset. Ground truth boxes are depicted in the left images, detections using Faster-RCNN with residual blocks in the middle and detections using LDS blocks in the right ones. Detections counted as True Positives are presented with green bounding boxes, whereas False positives (either due to wrong class label or low IoU value with the ground truth) are presented with red. To obtain these detections, score thresholds have been optimized independently for each block type by maximizing the sum of precision and recall.

3.3.3 Evaluation on object segmentation

Object segmentation is not yet a core module of surveillance video analytics due to technology limitations. However, DL has enabled robust object segmentation that can be important especially in complex and crowded scenes. We leverage the unique properties of LDS-ResNets to design and train a fully automatic segmentation method, aiming to provide robust segmentation accuracy, while reducing the required processing time. The proposed method incorporates distinct segmentation and error correction steps. Segmentation masks are produced by an independent segmentation model while erroneous labels are subsequently corrected by a combination of Replace and Refine networks [9].

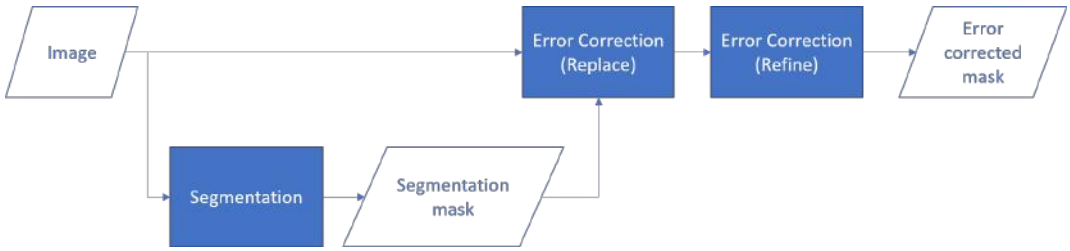


Figure 3.10: Flow diagram of the segmentation pipeline.

The core segmentation network consists of six consecutive Residual Blocks, depicted in Figure 3.4b. Residual networks have been proven to be easier to train, as shortcut connections help to deal with the problem of vanishing gradients. Since no fully connected layers are used, segmentation CNNs are fully convolutional, which makes their evaluation with different input sizes much more efficient [145]. In contrast to the U-Net architecture, [146, 147], the proposed segmentation architecture does not include any pooling layers and the spatial dimensions of the input image remain unaltered throughout the CNN. We argue that pooling operations suppress the input information into a more coarse representation which removes detailed information. With the exception of the last one, every convolutional layer is followed by a spatial batch normalization layer [137]. These layers improve the gradient flow, allow the usage of higher learning rates, minimize the effect of parameter initialization and act as a form of regularization. Batch normalization is skipped only after the last convolutional layer, since we do not want to alter the output's distribution. A ReLU activation [127] is added after most BatchNorm layers, which is defined as f(x) = max(0; x). ReLU activations do not limit the output's range, are computationally cheaper and lead to faster convergence rates during training. The last convolutional layer is followed by a $Tanh$ activation layer and subsequent addition and multiplication with constants to transfer the output's range to [0; 1].

The errors produced by automatic segmentation methods can be categorized to random and systematic [148]. While random errors can be caused by noise, systematic errors originate from the segmentation method itself and are repeated under specific conditions. Thus, systematic errors can be reduced using machine learning techniques. For example, a classifier may be built to identify the conditions under which systematic errors occur, estimate the probability of error for segmentations produced by a host method and correct the erroneous labels. CNNs have been successfully utilized for error correction purposes in the fields of human pose estimation [149], saliency detection [150] and semantic image segmentation [151].

There are two different alternatives regarding the output of error correction CNNs. Replace CNNs calculate new labels, which substitute the labels computed by the host method. On the contrary, Refine CNNs calculate residual correction values and their output is added to the host method's output. According to [148], each variant has its own shortcomings. Replace CNNs must learn to operate as unitary functions in case the initial labels are correct, which is challenging for deeper networks. Refine CNNs can more easily learn to output a zero value for correct initial labels, but face greater difficulties in calculating big residual correction in the case of hard mistakes.

The proposed method incorporates an error correction module targeted to systematic errors originating from the base segmentation algorithm. A Replace CNN is placed first, followed by the corresponding Refine CNN. The extended use of residual blocks in Replace CNNs minimizes the aforementioned problem of them having difficulties operating as unitary function when needed and lets them focus on correcting hard mistakes. With hard mistakes already corrected, Refine CNNs are used to make fine adjustments to the final labels. Thus, in the proposed architecture, we keep only the advantages of each error correction CNN variant and efficiently correct both hard and soft segmentation mistakes.

Every Replace and Refine CNN has the same structure with the Segmentation CNN, with two minor differences. The first is the number of channels in the input layer of error correction CNNs, which is equal to the number of inputs in each case (2 for Replace and 3 for Refine CNNs). Filter depth at the first convolutional layer is modified accordingly. The second difference only applies to the Refine CNNs. Layers after the last convolutional layer are omitted, since we do not need to explicitly restrict the output to a specific range in the case of residual corrections calculation. The Replace and Refine CNNs were trained in an end-to-end way, allowing their parameters to be co-adapted. The efficiency of the proposed method has been tested in [9], offering SoA performance in a fraction of the time required before for brain segmentation applications.

Concluding this section, a new ResNet-based CNN architecture inspired by Linear Dynamical Systems (LDS) was introduced, contributing towards the need to extract better features from surveillance video content. An extensive exploration of different network topologies has been presented to identify the optimal one. The proposed architecture has been validated in both generic and surveillance related tasks demonstrating improved performance in every task but also complementarity with other ResNet improvement methods.

Chapter 4

Consistent Optical Flow Estimation for Object Motion Analysis

The motion characteristics of each object/person in a scene observed by a surveillance system are essential to acquire a comprehensive understanding of the scene. The motion profile of the objects can be utilized to identify either their actions or broader events taking place. Moreover, motion information can be used in pre-processing operations (video stabilization) or in conjunction with appearance information as an extra modality to improve the performance of video analytics (object detection and classification). The optical flow of the scene is the preferred representation of this information modality. A literature review of optical flow estimation methods, with a stress on modern DL methods, is provided in Section 2.2.

This chapter presents the main contributions to the research community towards improving optical flow estimation. In the introduction, the intuition of the proposed method that uses a set of regularization methodologies to improve the quality of the flow without sacrificing performance is described. Subsequently, the proposed architecture is fully described and validated in popular publicly available datasets.

4.1 Introduction

The computation of optical flow is an ill-posed problem in its naive formulation, as there are multiple valid solutions. Training a Deep Neural Network (DNN) network for a complex problem is generally a cumbersome procedure, highly sensitive to the training set and the parameters used. In the relevant literature, different strategies have been followed to improve accuracy by either modifying the objective function or controlling the biases, a.k.a. regularization methods. Regularization can be defined as any strategy employed to improve the training procedure of a neural network by imposing problem-specific restrictions. It has been shown that regularization can improve the behavior of DNNs [152], mitigating the inevitable bias of the training set and guiding them towards more generalized solutions.

Motion consistency is a well-established principle used for the regularization of optical flow estimation networks. Neighboring points of the scene are assumed to move uniformly; a hypothesis that is accurate in most cases. However, this assumption fails significantly on object boundaries, as depicted in Figure 4.1, as well as on joints of non-rigid objects. Consequently, methods that adopt the motion consistency principle tend to suffer from "soft"and inaccurate edges between objects. This phenomenon is more evident when two neighboring objects are moving in opposing directions. Edge maps have already been used in optical flow estimation [153] in order to enable special computational strategies on the edges. However, these methods still rely on low-level edge extraction and often fail to distinguish possible motion-separating edges from simple texture patterns or variations. We show that semantically richer information can be used to improve flow estimation accuracy and motion consistency.

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

Figure 4.1: Illustration of global smoothing deficiency. (a) Ground truth optical flow. (b) Flow from a globally regularized network. (c) Flow from a locally regularized network. Notice that the silhouette in (c) is visibly less blurry than in (b).

Optical flow and semantic segmentation may be seen as disparate modalities given that the former is related to motion information and the latter is related to scene understanding. However, the optical flow vectors of a natural scene are constrained by the underlying physics of the world, while the semantics of an object are closely related to the patterns of its motion. The close relationship of the aforementioned modalities is confirmed in the literature, as they are jointly exploited [154, 57]. The affinity of the two research domains is demonstrated in [155] and further grounded in recent neuroscience reports such as [156, 157].

An explicit regularization method is presented in the current work to guide any optical flow estimation network to better model the correlation between motion flow and texture on the edge of an object. The training procedure is regularized by selectively imposing the motion consistency constraint on semantically coherent local regions of the image. For this purpose, a new loss function is introduced that takes into account the semantic segmentation mask of the examined scene to reason on the validity of motion consistency. Based on this approach, the network learns to implicitly distinguish inter- and intra-object edges and to reason differently on the motion field. At the inference stage, the network produces sharper, more consistent flow within each object and less motion field leaks across objects without requiring any additional input or modification.

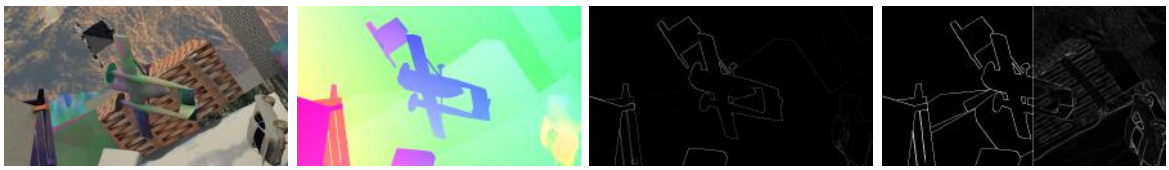
An implicit regularization method is also proposed in this work for optical flow estimation networks, aiming to improve the training procedure by reducing the legitimate solution space. DNNs are designed to be translation invariant in order to improve their ability to detect textures and objects within an image. This translation invariance stems mainly from the pooling layers of the network, while convolutional layers are also contributing to it. While in some applications this is a desirable property, a recent work [158] explored its potential deficiencies for different types of computer vision applications. It is argued here that for optical flow estimation, translation invariance introduces learning perturbations impeding the efficient training of the network. Thus, by removing the translation invariance, the training procedure is implicitly regularized, leading to a faster convergence with increased accuracy.

Moreover, an exploration of both regularization strategies, explicit and implicit, as well as their complementarity, is performed in order to better understand the role of regularization in optical flow estimation. For this purpose, the widely adopted flow estimation network FlowNet2 [12] and its building blocks are used as a baseline. An exploration study is performed to verify the validity of our claims, gradually integrating the regularization strategies and building up in network complexity. The evaluation of the proposed methods against the baseline is performed on standard datasets used as a benchmark for optical flow estimation.

4.2 Regularization for improved optical flow estimation

As already mentioned, regularization can be defined as any strategy employed to improve the training procedure of a neural network and to reduce its generalization error. In the relevant literature, different strategies have been followed, imposing restrictions and penalties that can lead to improved performance by either modifying the objective function or controlling the biases. Regularization techniques can pursue their aim among others by (1) encoding specific prior knowledge, (2) promoting a simpler solution, (3) transforming the problem into a determined one or (4) combining multiple hypotheses to model the training data. In this work, we are mainly focusing on (1), exploring the use of semantically driven local motion consistency in a semantic context, and (3), reducing the spatial invariance of the network to better determine the solution space. The exploration process performed for the proposed approaches is also presented, as it can offer significant insights in the discussion of the results.

4.2.1 Semantically driven local motion consistency



(a) **(b)** **(c)** **(d)**

Figure 4.2: (a) Sample from FlyingThings3D dataset. (b) Ground truth optical flow. (c) Motion edges. Notice how the lion's share of variation is concentrated on object boundaries. (d) Comparison of semantic edges with a typical intensity edge mask. Please, notice how much better the semantic edges correlate to the motion ones.

Explicit regularization techniques impose some constraints directly on the output of the estimator, which in the case of deep neural networks takes the form of a loss term. The simplest method is global consistency (a.k.a. global total variation), under which large, abrupt changes of the predicted motion field are penalized. Naturally, such a penalty decreases the accuracy in regions where the transition of the ground truth flow is sharp, such as on the boundary of two independently moving objects, as depicted in Figure 4.1. This becomes particularly evident in cases of objects moving in the opposite direction at a different depth.

In such cases, a local regularization term that selectively regularizes motion consistent regions is required. These regions are bound by edges with sharp motion change (a.k.a. motion edges). The identification of motion edges is performed by utilizing low level features such as intensity edges and gradients in their modeling [37], based on the observation that independently moving objects have distinct appearance and can be separated from an intensity edge. As it can be seen in Figure 4.2(d), though, such features are noisy motion edge predictors since arbitrary shape and texture exists within rigidly moving entities. Semantic segmentation, on the other hand, encompasses meaningful information about the rigidity and the actual boundaries of the objects. Figure 4.2 shows that edges from a semantic segmentation mask are strongly related to motion edges and consequently we propose the use of such semantic edges as guides for our local, semantically driven local smoothing. A more elaborated example can be found in Figure 4.18.

Given any optical flow estimation network, we modify the training branch by adding to the EPE loss the proposed Local Smoothing Loss, as depicted in Figure 4.3. No other modifications are performed to the architecture of the baseline network. Therefore, the inference procedure is not affected in terms of complexity or time. The added loss term penalizes motion inconsistency inside an object, guided by the semantic segmentation mask, while ignoring areas on the boundaries with other objects. The aim of this approach is to regularize the network, driving it to learn whether the image texture contains semantically important edges signaling motion images, rather than intensity edges. Proper generalization is important as the network should discriminate them without a segmentation mask at the inference stage.

The loss is implemented as an element-wise multiplication of the consistency term with a binary mask that contains the edges extracted from the semantic instance segmentation. Motion spikes filtered by the mask are excluded from the consistency constraint and are, thus, not regularized. Let us define the semantically-driven local motion consistency loss as:

$$l_{smooth} = \frac{\sum((V_x + V_y) \odot M_{sem})}{\sum M_{sem}} \tag{4.1}$$

with V_k, where $k \in \{x, y\}$, being:

$$V_k(i,j) = |k_{i+1,j} - k_{i,j}| + |k_{i,j+1} - k_{i,j}| \tag{4.2}$$

i and j refer to the rows and columns of the optical flow field, V_x and V_y are the total variations for the two optical flow channels, M_{sem} is the smooth patch boundary binary mask and $\odot$

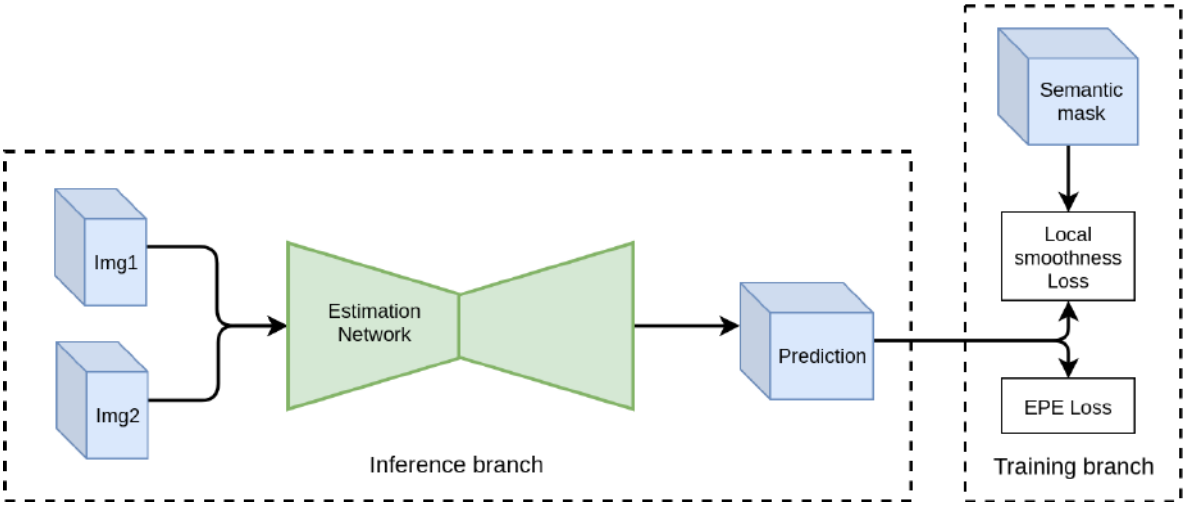


Figure 4.3: Illustration of the proposed training scheme. The additional loss component implicitly regularizes the predictions during training time, requiring no modifications at inference.

indicates the Hadamard product. The total loss for our training procedure becomes:

$$l_{total} = l_{AEE} + \alpha * l_{smooth} \tag{4.3}$$

with alpha regulating the contribution of the semantically driven consistency term and l_{AEE} the average endpoint error loss term. The endpoint error can be calculated as $\|V_{pred} - V_{gt}\|$ with V_{pred} being the predicted optical flow and V_{gt} the ground truth vector.

M_{sem} can be calculated by applying the derivative operator to an object instance segmentation mask, resulting in a binary mask that delineates object boundaries, as can be seen in Figure 4.18c. While ground truth object instance segmentation information is available for many flow datasets, some misalignment between the different modalities can exist, as it can be seen in Figure 4.4. In order to ensure good alignment between M_{sem} and ground truth flow spikes, we apply a dilation operator after the derivation. A 3×3 dilation kernel proved sufficient, as almost all misalignment is up to two pixels.

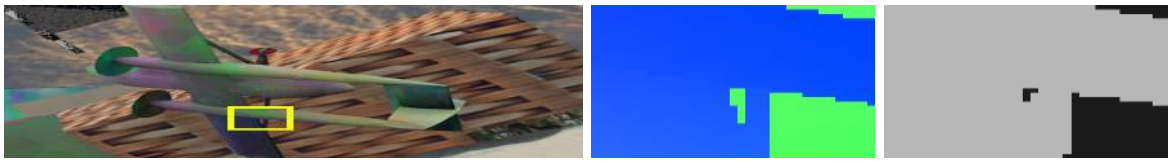


Figure 4.4: Sample from FlyingThings3D dataset with a cutout of its corresponding ground truth optical flow and object segmentation. Notice how the transition from the airplane to the background object happens at different positions in the two modalities.

4.2.2 Coordinates as regularizing features

The idea of integrating pixel coordinates as features in CNN architectures was explored by Liu et al. [158]. They investigated the merit of utilizing pixel grid coordinates as input features at various stages of a CNN, introducing the CoordConv module (Figure 4.5). In this way, the output is being determined not solely by content-based feature values, but also from the position of those features on the grid, breaking the celebrated translation invariance property. In theory, the network can nullify the effect, learning to discard these features if necessary.

The methodology was tested on the toy problem of mapping coordinates in (x, y) Cartesian space to coordinates in an one-hot pixel space, significantly improving performance and convergence. While it seems intuitive to utilize coordinates as inputs on such a task, the expected effect is heavily domain dependent and should be evaluated on a task by task basis. In image classification, for example, the impact of CoordConv is not significant, while in object detection, generative modeling and super-resolution [159], important improvements are noticed. The selection of the convolutional layers to be modified is also important.

We claim that the accurate pixel correspondence calculation required in optical flow estimation inherently relies on such a feature. To assess its merit, we test the impact of CoordConv on a toy optical flow estimation problem. We construct a simple dataset, containing 3136, 32×32 image pairs with 9×9 moving rectangles and their corresponding ground truth optical flow (Figure 4.7). Each rectangle can freely move within the image following a uniform distribution.

We train a simple autoencoder based network, depicted in Figure 4.8, on this simple dataset, with and without CoordConv layers and compare their validation performance. The implementation of the CoordConv model is simple and straightforward. We concatenate a two-channel coordinate tensor at the input of the first and last layer. The coordinates are centered around zero and normalized in the range $[-1, 1]$. A trivial example of unnormalized

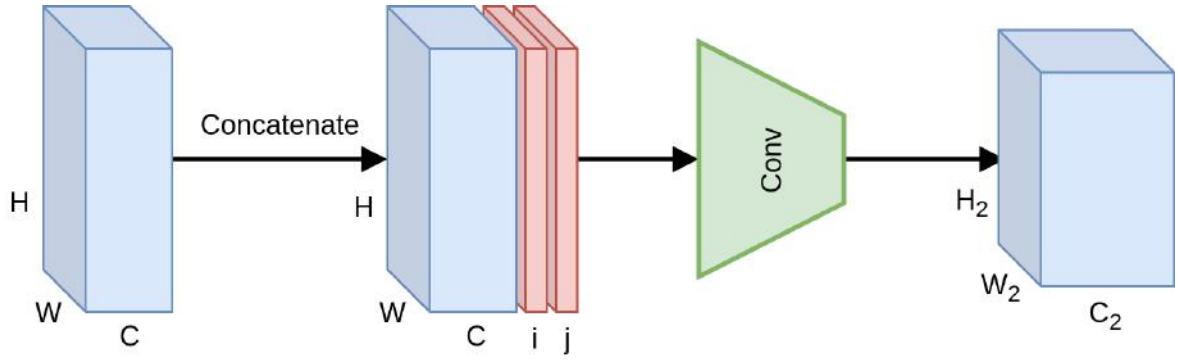


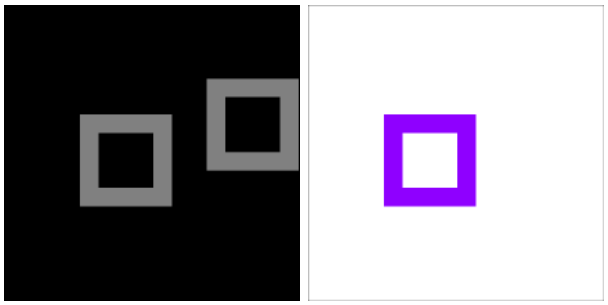
Figure 4.5: Schematic representation of the CoordConv module.

coordinates for a 5×5 image is depicted in Figure 4.6.

-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2

Figure 4.6: Example of an unnormalized x-coordinate tensor for a 5×5 pixels image.

The training of the network using CoordConv converges after 85 epochs, in contrast to the baseline network which fails to converge even after 175 epochs (Figure 4.9). The experiment was repeated 100 times for validation. It is evident that training is more stable using CoordConv, with the majority of the experiments converging around the same error value, as indicated by the low standard deviation of the error curve. In Figure 4.10 the improvement is clearly depicted from the qualitative comparison of the example flows. While the baseline network sometimes outputs fuzzy and inconsistent outlines, using CoordConv the output consistently follows the shape of the ground truth. This performance gap cannot be attributed to suboptimal hyperparameter values, with the relative difference between the two models remaining the same across a wide range of learning rates.



(a) (b)

Figure 4.7: (a): Overlay of the images from sample of our toy dataset. (b): Ground truth optical flow.

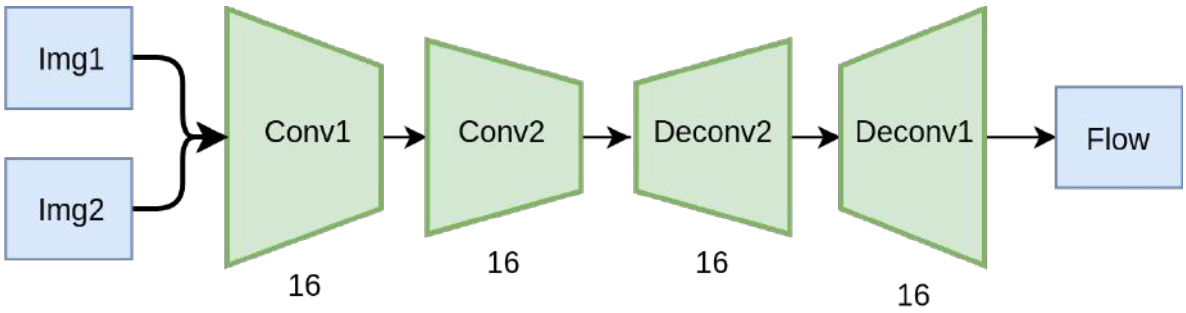


Figure 4.8: Overview of our baseline simple architecture. On the modified version CoordConv modules are appended to layers Conv1 and Deconv1.

4.3 Experimental Evaluation

After demonstrating the theoretical merit of the proposed regularization techniques, in this section they are extensively evaluated.FlowNet2 [12] has been selected as the reference architecture for the experimental evaluation, since its modularity allows for easier training, achieving high performance and accuracy. It comprises a stack of backbone modules, namely FlowNet C and FlowNet S which implement the well known encoder-decoder scheme. Together, they form a refining stack, since each module takes as input the previously computed flow and improves upon it. Overall, each added module takes as input the two reference images ($Img1$ and $Img2$), the previously estimated flow, the warped image $Img2_{warp}$ and the

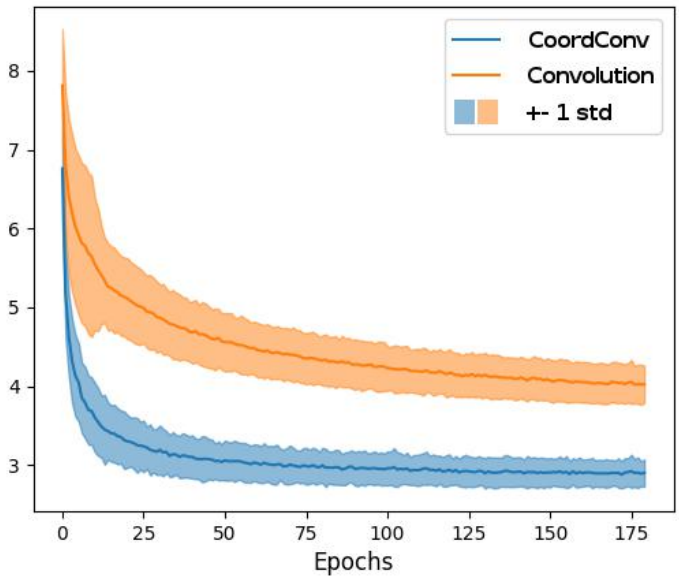


Figure 4.9: Magnitude and standard deviation of endpoint error for our simple networks trained on our toy dataset. Notice that the network with the CoordConv module converges significantly faster and on lower error values.

error between $Img2_{warp}$ and $Img1$. $Img2_{warp}$ is computed using backward bilinear warping on $Img2$ using the flow field from the previous stage. As the network stack gets deeper and the predicted flow gets progressively refined, approaching the ground truth, $Img2_{warp}$ should approximate $Img1$ and the error reaches zero.

In order to evaluate our contributions, we apply the proposed regularization methods to FlowNet S and proceed with training of the network, replicating the gradual training procedure of FlowNet2. The modified FlowNet S module is constructed as depicted in Figure 4.12. The evaluation is performed in two stages, as depicted in Figure 4.11. At the first stage, a stack comprising a FlowNet C module, followed by a FlowNet S is created, forming FlowNet2 CS. The contribution of the regularization based on a) the semantically-driven motion consistency and b) the coordinate feature layer, is studied comparing the baseline FlowNet S module and the proposed modified one. At the second stage, another FlowNet S module is added to the stack forming a FlowNet2 CSS and the proposed regularization methods are

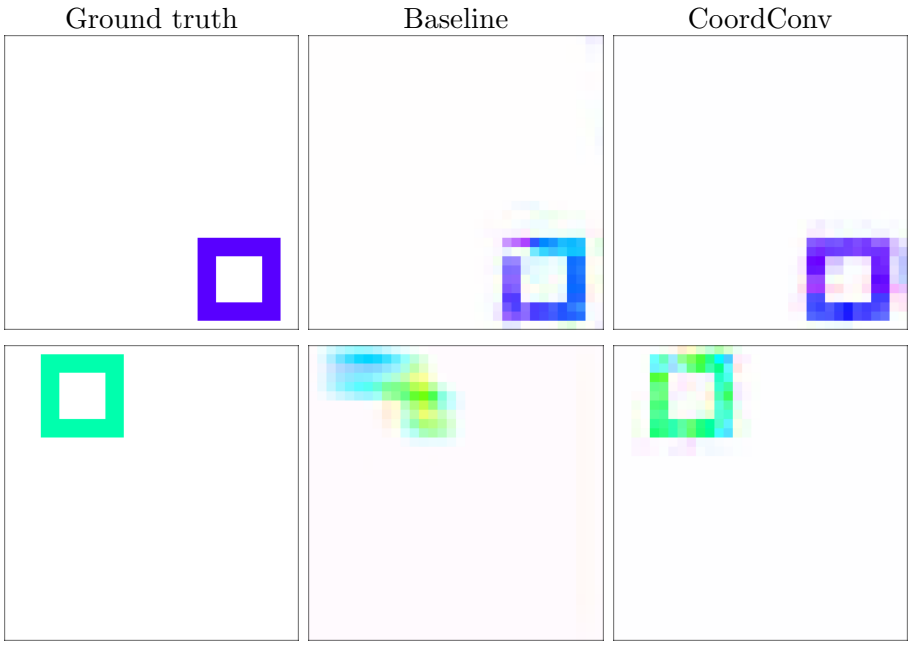


Figure 4.10: Validation samples after convergence on our toy dataset.

again assessed, both quantitatively and qualitatively, to prove the validity of our claims.

4.3.1 Evaluation framework

The FlowNet2 building blocks were implemented in Pytorch, namely FlowNet C and S. Due to the fact that we cannot reproduce accurately the augmentation process followed in the original work, we demonstrate results comparing our implementation and the modified version proposed. Both networks are trained identically, using the same train/test split ratio, training schedule (epochs) and learning rate.

The training of the models is performed using FlyingChairs [38], containing 22872 samples, and FlyingThings3D [160], containing 80604 samples. Although both of them are synthetic datasets that contain unrealistic scenes, they are widely used as training sets in optical flow estimation, since they contain a large number of samples, leading to good generalization.

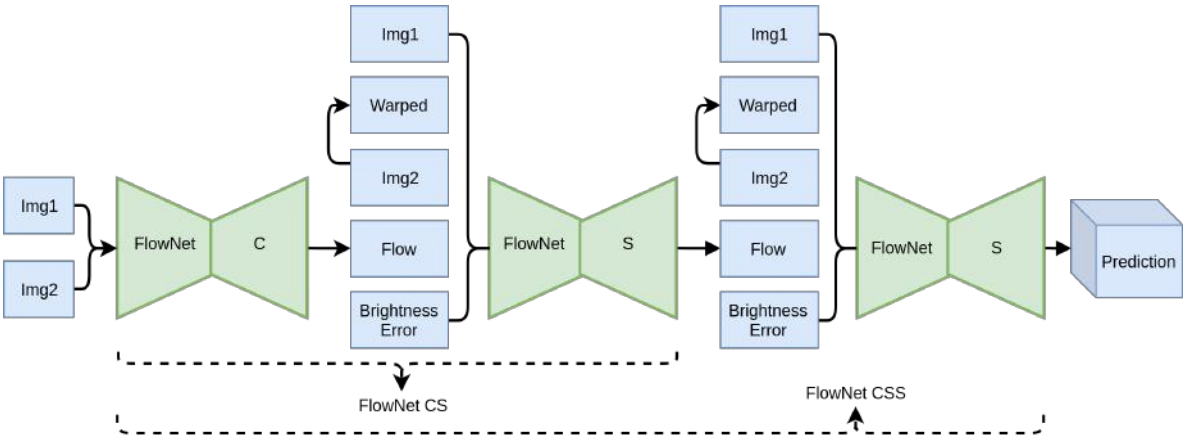


Figure 4.11: Overview of the network architecture. We utilize three stacked autoencoder refinement modules.

Throughout the training process, we assume 80–20 train and test splits and use reference splits where provided.

Following the procedure proposed in the original implementation, each network is firstly trained on FlyingChairs (Long scheduling) and then on FlyingThings3D (Fine scheduling) for a total of 1.7 million iterations. Under the long schedule, the learning rate is halved every $200k$ iterations after the $400k$ mark and until reaching $1.2M$ iterations. Fine scheduling covers the last $500k$ iterations and the learning rate is, again, halved every $100k$ iterations after the $200k$ mark.

The evaluation is conducted on MPI Sintel [161], KITTI datasets [162, 163], and Middlebury [164]. MPI Sintel is synthetic but contains plenty of dynamic scenes (1041 samples) with complex motion and photo-realistic effects. KITTI datasets are the sole available large-scale, realistic ones, but they contain only 394 samples in total and the available ground truth is sparse. Finally, Middlebury consists of 4 synthetic and 4 natural samples with dense ground truth. The metrics used for evaluation are the widely adopted average flow vector endpoint error (AEE) and the outlier percentage (OP). For the latter, a pixel is considered to be an inlier if its endpoint error is $< 3px$ or $< 5\%$.

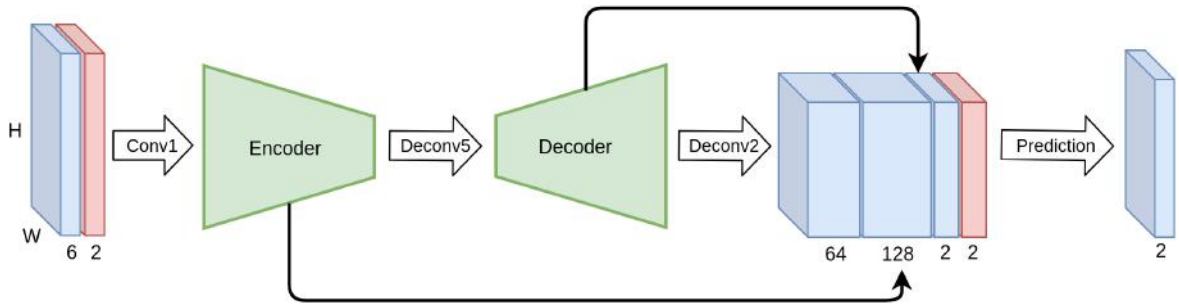


Figure 4.12: Overview of the modified FlowNet S architecture with added CoordConv modules. Grid coordinates (red) are concatenated before the first and last convolution.

4.3.2 Exploration study

In this section, we perform an exploration study, independently testing our contributions on a smaller scale. We train FlowNet2 CS on FlyingChairs using different types of regularization.

Initially, the effectiveness of all consistency-based regularization schemes is evaluated. "No regularization" is considered as the baseline and it is compared against the use of global and semantically driven local consistency as regularization constraints. Following the literature, the regularization factor a in (4.3) is set to 0.01 for all the experiments performed. This value leads to a relative weight of the regularization term to the main training objective of $\approx 1\%$. The results in terms of AEE and OP are depicted in Table 4.1.

Table 4.1: Global vs local smooth results trained on FlyingChairs with long scheduling.

	Sintel clean		KITTI 2012		KITTI 2015	
Method	AEE	OP	AEE	OP	AEE	OP
FlowNet2 CS	2.89	**12.15**	8.07	41.22	16.05	48.53
Global Smooth CS	2.84	12.40	7.29	37.10	15.74	46.88
Local Smooth CS	**2.82**	12.37	**7.23**	**36.58**	**15.39**	**46.00**

The results show that methods using regularization based on both the global and local consistency loss are surpassing the baseline on 2 of the 3 evaluation datasets. The proposed semantically driven local consistency loss leads to the best results on KITTI 2012 and 2015, both in terms of AEE and OP. On Sintel, both regularization methods are producing a lower

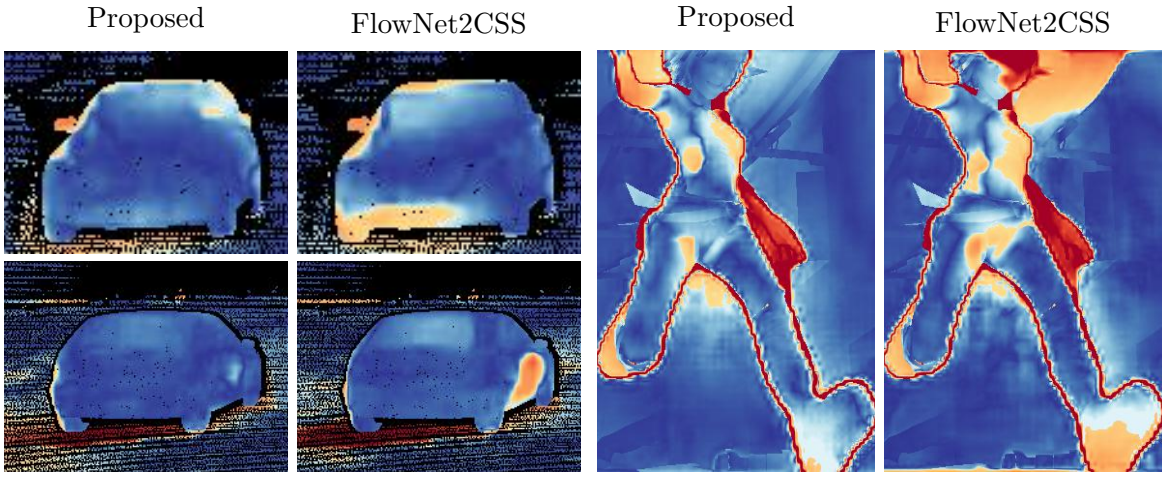


Figure 4.13: Closeup at object boundaries between the proposed method and FlowNet2CSS.

AEE than the baseline but a higher Outlier Percentage (OP). This is attributed to the fact that Sintel has similar characteristics to the training set and small non-regularized networks tend to overfit to the training data. The use of the semantically driven local consistency loss produces results superior to the global one in all cases. As it is validated in the next sections, fine tuning the network with a dataset rich in semantic entities generalizes better and the regularized network has better performance even in simple datasets. Similarly, networks with a bigger capacity perform better at all cases. Overall, regularizing the network using consistency constraints improves the optical flow estimation over the baseline.

The qualitative difference between the two methods, is highlighted in Figure 4.1. It is evident that the flow on the silhouette of the object is crispier with visibly less blur when using the semantically driven local consistency loss, in contrast to the over-smoothed boundaries of the globally regularized example. In complex scenes that involve many objects, the result is visibly better combining sharp edges and smooth flow within the objects.

After evaluating the effect of our novel consistency constraint as a regularizer, we move forward to incorporate the proposed modification of FlowNet S with the CoordConv layers. FlowNet2 CS and the proposed modified network are trained following the procedure described in Section 4.3.1, performing both the long and the fine scheduling training stages. The results for each stage are reported in Table 4.2.

Table 4.2: FlowNet2 CS results vs baseline on different training steps.

Method	Sintel AEE	Sintel OP	KITTI 2012 AEE	KITTI 2012 OP	KITTI 2015 AEE	KITTI 2015 OP
Long scheduling on FlyingChairs						
FlowNet2 CS	2.89	12.15	8.07	41.22	16.05	48.53
Proposed CS	**2.80**	**10.31**	**6.76**	**31.10**	**15.08**	**44.06**
Fine scheduling on FlyingThings3D						
FlowNet2 CS	2.27	8.42	4.96	21.43	12.11	34.22
Proposed CS	**2.22**	**8.12**	**4.42**	**20.20**	**11.15**	**33.00**

The modified FlowNet2 CS architecture benefits from our contributions, outperforming the baseline both in terms of AEE and OP in all used datasets. The improvement is consistent in both training stages and is more evident in the realistic KITTI 2012 and 2015 datasets, where the proposed network improves the baseline by 0.5 to 1 pixel in terms of AEE and by 1 to 2.5 points in terms of OP. Moreover, comparing Table 4.1 and Table 4.2, the additive improvement of the two methods can be seen, validating their complementarity.

4.3.3 Performance evaluation

In the previous section, it was shown that regularization, implicit or explicit, has significant added value in an existing optical flow estimation model. However, stacking flow estimation modules is also effectively acting as a flow regularizer, improving the overall performance. In this section, the effect of the proposed regularization on a bigger stack is examined.

The proposed modified FlowNet S module is attached to the CS stack forming FlowNet2 CSS. With the weights of first stage fixed, training is continued on FlowNet2 CSS, following the same procedure. Both a long and a fine training schedule are employed and the results are shown in Table 4.3.

Quantitatively, the proposed CSS network outperforms the baseline FlowNet2 CSS on all available datasets by a significant margin. KITTI sets are the most representative ones of a real world scenario and constitute the de facto test for modern algorithms, while Middlebury is a renowned dataset with real samples. It is worth noticing that the results of the proposed

Table 4.3: FlowNet2 CSS results vs baseline after the final training step.

Dataset		FlowNet2 CSS	Proposed CSS
Middlebury	AEE	0.55	**0.41**
	OP	2.15	**2.07**
Sintel clean	AEE	2.12	**2.02**
	OP	7.61	**7.02**
KITTI 2012	AEE	4.69	**4.01**
	OP	19.50	**17.76**
KITTI 2015	AEE	11.89	**10.66**
	OP	31.87	**30.52**

CSS are lacking in performance when trained only with the simple FlyingChairs dataset, while training on FlyingThings3D is drastically improving the results. Despite the fact that stacking is implicitly regularizing the network, the proposed approach is further improving the results increasing the lead from the baseline in the KITTI datasets.

Upon qualitative inspection of the results as depicted in Figure 4.15 and 4.16, the numerical improvement is clearly reflected on the results. It is clear that the proposed network handles the inherent motion consistency of planes (e.g. roads and pavements) and objects better than the baseline. In Figure 4.17, where scenes from Sintel are depicted, the results of the proposed network maintain better sharpness on object boundaries. The objects are delineated in a more clear and consistent manner.

Shadows constitute a particularly interesting case for optical flow algorithms, because they break the brightness constancy constraint. More specifically, unsupervised methods that are trained with reconstruction loss, produce erroneous flows at shadows, since they appear to move together with the object casting them. The proposed method enforces flow consistency within semantic entities, reducing the errors in those areas. Examples of this behavior can be seen in Figure 4.14.

Figure 4.14: Typical examples of improved shadow handling with our approach. Shadows that appear to move with their casting objects are better mapped to the background. Error magnitude increases transitioning from dark blue to yellow and red.

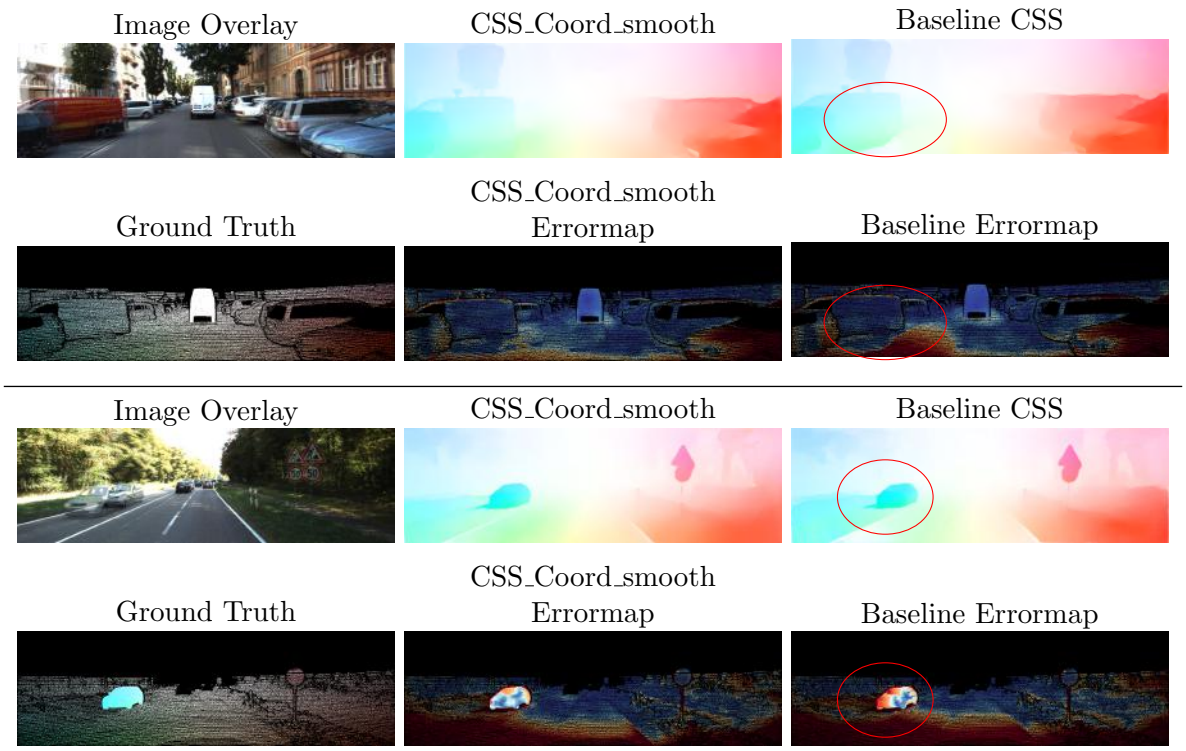


Figure 4.15: Qualitative results on the KITTI2015 dataset. The network with our contributions produces more accurate, detailed and smooth results than that of the baseline network.

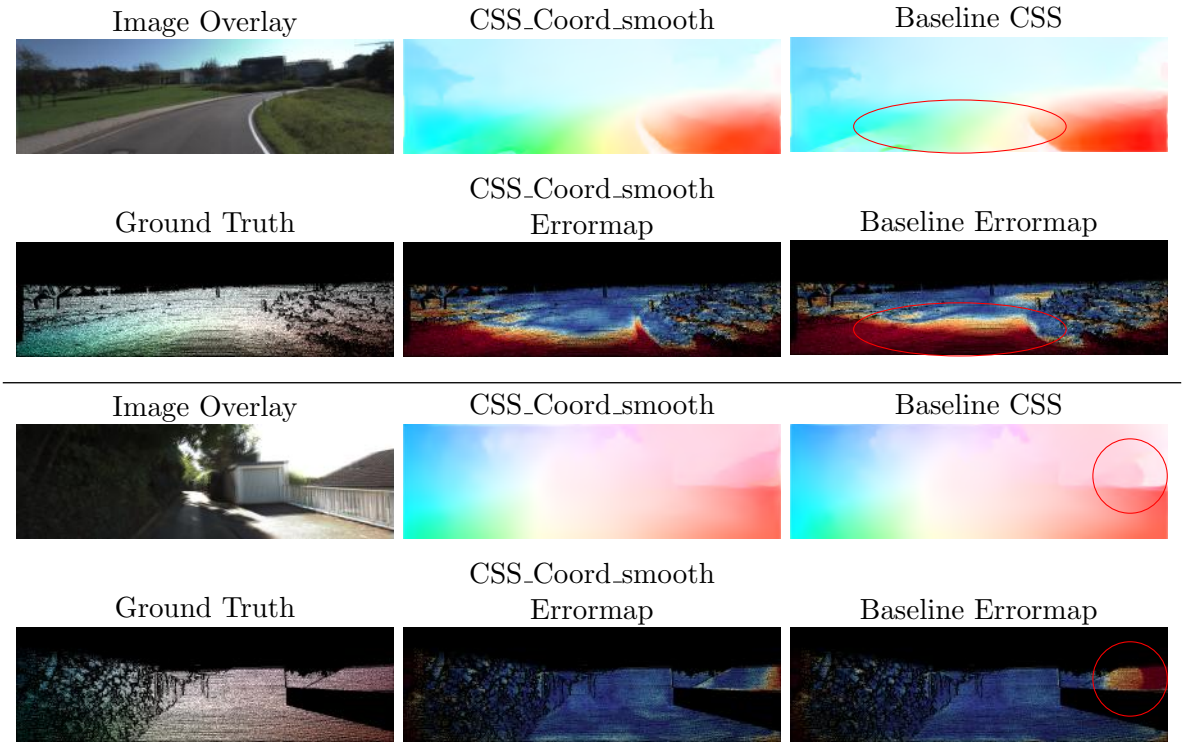


Figure 4.16: Qualitative results on the KITTI 2012 dataset. The network with our contributions produces more accurate, detailed and smooth results than that of the baseline network.

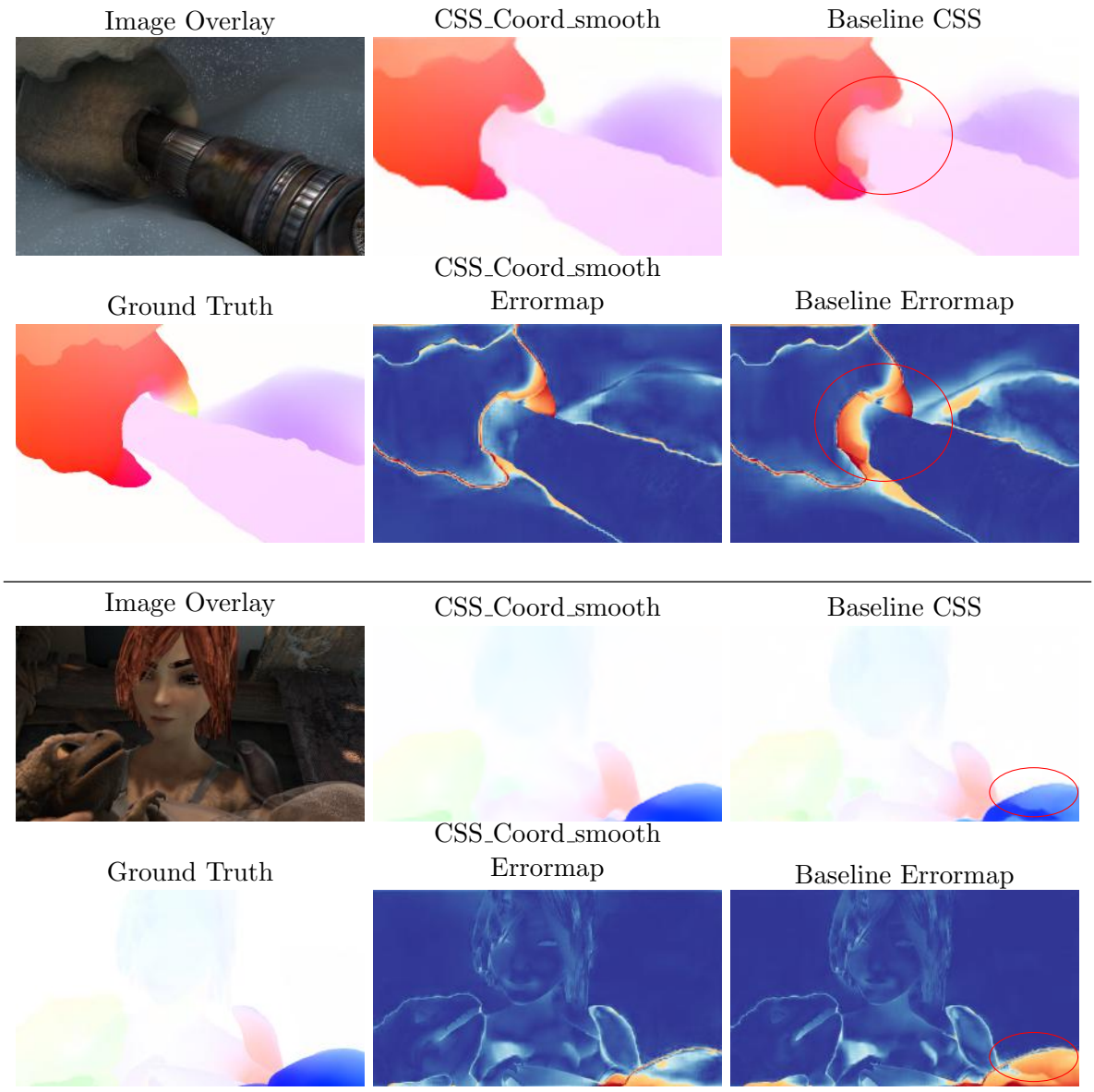


Figure 4.17: Qualitative results on MPI Sintel dataset. Although the baseline network overfits and presents lower error magnitudes on this synthetic dataset, our method maintains its qualitative edge, producing flow vectors that adhere to object boundaries.

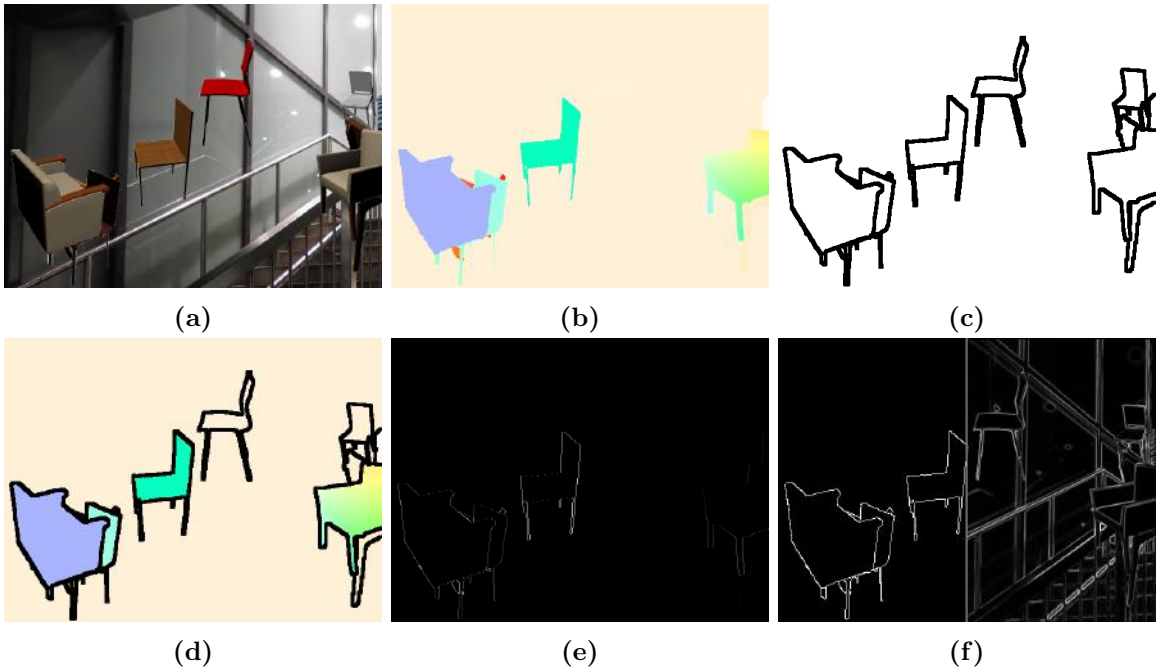


Figure 4.18: (a) Sample from FlyingChairs dataset. (b) Ground truth optical flow. (c) Corresponding M_{sem} mask. (d) Overlay of M_{sem} and GT flow. Notice how enclosed segments have a smooth flow. (e) Motion edges. Notice that it is concentrated on boundaries. (f) Comparison of semantic edges with a typical edge mask. Notice how they better correlate to motion edges.

Concluding this section, two regularization methods have been proposed to improve optical flow estimation. Initially, a new loss function for explicit regularization is proposed that takes into account the semantic segmentation mask of the examined scene, enforcing motion consistency within the object and sharpness on the boundaries. Different loss designs are extensively explored and tested at multiple stages of the baseline architecture. Moreover, the use of pixel coordinate information as an implicit regularizer in optical flow estimation networks is proposed. An in depth exploration of the role of translational invariance on the quality of flow estimation is performed to validate our claims. The proposed regularization methods along with their complementarity, have been extensively assessed on the optical flow estimation task in popular publicly available datasets and insights on the training procedure are provided. Even though the improved flow is not directly contributing to video analytics, it is a building block for enhanced object detection (Section 5) and video stabilization (Section 7).

Chapter 5

Fast and Robust Object Recognition

Object recognition is a fundamental tool for surveillance applications, providing semantically meaningful information about the content of the footage examined. DL techniques have vastly improved the capabilities of computer vision to detect and classify objects. Superhuman vision capabilities have been claimed in the framework of specific image recognition competitions. However, there is still room for improvement, especially in uncontrolled environments faced by surveillance cameras, featuring bad lighting, low resolution, Pan-Tilt-Zoom (PTZ) operations, heavy occlusions, clutter, and complex background [165, 166].

Object recognition is often broken into two stages: detection and classification. In the first stage, objects are located in the scene and, subsequently, they are classified into a semantic class (e.g. person). A literature review of object recognition methods, with a stress on modern DL methods, is provided in Section 2.3.

This chapter presents the main contributions to the research community towards improving object recognition. Two separate lines of work are presented aiming to improve SoA methods by dynamically parameterizing the object detection step to better adjust to the appearance evolution of the object tracked, and by taking into account the behavior pattern of the objects to enhance the classification step. The proposed methods are described in detail and validated in publicly available datasets.

5.1 Improved object detection for surveillance applications

In this section, a multiple-object detection framework that confronts the challenges of real surveillance footage is proposed. It is based on a state-of-the-art object detection and localization framework [63] that is trained offline to facilitate a tracking-by-detection paradigm. A novel methodology is proposed to dynamically control the detector configuration using estimations of the intrinsic parameters of the camera. A Recurrent Neural Network (RNN) is employed to model the spatial transformation [167] of the objects due to the camera perspective and PTZ operations of the camera. The RNN is used to predict the affine transformation of the objects in the subsequent frames and dynamically modify the parameters of the detector. Moreover, application-specific training data augmentation is examined to streamline the performance of the detector. Experimental validation of the proposed concept is performed on real CCTV videos, with a focus on pedestrians. Nevertheless, the proposed framework is applicable to any type of objects.

5.1.1 Dynamic detector configuration

CCTV cameras often have PTZ capabilities that are used by their operators to track suspicious activities in a scene. These camera operations constitute a serious challenge for object detection and tracking methods due to the implicit scale assumptions made. Object detection techniques have a predefined range of scales that are supported, to minimize detection errors. In this section, it is proposed to dynamically adjust this scale range based on predictions of the tracked objects' size in the next frame.

The first step towards this approach is to have an accurate estimation of the detected object's scale and pose. The Spatial Transformer Network (STN), proposed in [167], can apply a spatial transformation operation to a feature map during a single forward pass. It can be inserted as a layer in a feed-forward convolutional network. It learns an affine transformation of the input and uses bilinear interpolation to produce its output allowing it to zoom, rotate and skew the input. The transformation parameters (5.1) can be also exploited as a robust indication of the object's scale and pose.

$$A = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \tag{5.1}$$

An RNN is subsequently used to model and predict the evolution of the transformation matrices in the next frame. The transformation matrix of the detected objects is input to the recurrent network such that:

$$A_t = STN(f_{conv}(I)) \tag{5.2}$$

$$h_t = f_{trans}^{rnn}(A_t, h_{t-1}) \tag{5.3}$$

where A is the transformation matrix from the current object, STN is the spatial transformer module and h_{t-1} is the hidden state of the RNN model in the previous step. An affine transformation matrix A_{t+1} is produced at each time-step t from the hidden state of the RNN. The affine transformations predicted are conditioned on the previous transformations through the time dependency of the RNN.

The produced A_{t+1} is utilized to dynamically find the optimal scale range for the detector, in this work Faster R-CNN. To achieve that, we modify the scaling hyperparameter s of Faster R-CNN that controls the scale of the processed image, achieving better scale invariance. The proposed approach is experimentally validated with real surveillance footage in Section 5.1.3.

5.1.2 Training data augmentation

CCTV videos often contain severely blurred objects due to low video quality and fast PTZ operations. Especially motion blur is a major challenge for object detection methods in CCTV content. In a single frame, motion blur is translated to degraded appearance information and reduced ability to accurately localize the position of an object. While de-blurring methodologies show good results [168], they have a high computational cost and they further degrade image quality. Deep learning methods have been recently shown to achieve impressive performance in benchmark datasets for object detection. However, in challenging CCTV videos their performance deteriorates. In this section, the effect of training data selection in the detector's performance is explored.

Building on prior deep learning work, the object detection and localization framework Faster R-CNN [63] is employed. For this work, a ZF network model [169], pre-trained in ImageNet dataset [18], is selected as the backbone network for object detection. The model is fine-tuned to optimize the discriminative power of the features learned and, therefore, the detection

accuracy. Fine-tuning has been widely used in DL, greatly improving the performance of CNNs. It has been shown [170] that transfer learning, namely the use of pre-training in a generic dataset, has significant value offering a robust initialization of the network parameters.

Two training data augmentation approaches are examined to improve fine-tuning of the examined system: (a) the enrichment with object instances from heterogeneous sources and (b) the addition of blurred instances of the current object collection. In the former approach, annotated datasets featuring the examined object classes are utilized. An extended training set is created that contains samples from multiple datasets.

Despite the existence of several annotated datasets, their content is produced with quality measures that are far superior to the conditions that a normal CCTV system will face. Therefore, the features learned by a deep learning object detector are often plagued by many missing detections, especially in action scenes. Following the latter approach, the training set is augmented with blurred instances to enhance the robustness to motion blur. A set of Gaussian kernels incorporating motion blur [171] has been created (5.4).

Figure 5.1: Examples of motion blur effect on images.

$$\mathbf{K} = \{k_{\theta,l} | \theta \in \Theta, l \in \mathbf{L}\} \tag{5.4}$$

As the kernel k is symmetric, the motion direction θ is randomly sampled from $\Theta = [0, \pi]$ and the magnitude is selected from $\mathbf{L} = [0, l_{max}]$, where l_{max} is a parameter. The original images I that form the training set are convolved with the motion-blur kernels.

$$\mathbf{I}_{bl} = \mathbf{I} \otimes \mathbf{k} \tag{5.5}$$

Figure 5.1 shows examples of using kernels to generate blurred images with different parameters. The proposed data augmentation is experimentally validated in Section 5.1.3.

5.1.3 Experimental Evaluation

In this section, the experimental setup for the validation of the above concepts is being described. Given that pedestrians is the main object class of interest, the experiments will focus on the pedestrian detection without losing generality. For this purpose, a number of publicly available datasets have been selected to feature the experiments. VOC2007 [172] is used as a generic dataset for image classification with 20 annotated classes, including the class *person*. The ETH dataset [173] is also used to extend the fine-tuning dataset. It contains annotated pedestrians on a public road. Finally, a set of videos from the Metropolitan Police of London (MET) from the riots of 2011 have been also used for qualitative validation. Those videos have been offered for research purposes in the framework of the LASIE FP7 EU funded project and they are neither annotated nor publicly available.

The first set of experiments refers to the exploration of training data augmentation strategies. The Faster R-CNN object detection framework is fine-tuned with different training sets to test their effectiveness. Training with VOC2007 (∼10000 object instances), labeled as [VOC], is used as a baseline for performance. The training set is infused with sequences from the ETH dataset, namely "Bahnhof" sequence (∼7500 object instances) labeled as [BAH] and "Sunny Day" (∼1900 object instances), labeled as [SUN]. The datasets are divided in training and testing set of equal size. 50% of the training set is used for validation purposes. The evaluation of the trained models is performed on separate testing sets that include VOC testing set and an ensemble of the VOC and ETH testing sets, respectively. The results are reported in Table 5.1. Experiments show that the detection accuracy seems to benefit from extra training samples, even on the original VOC testing set.

	VOC	VOC+BAH	VOC+SUN
[VOC]	59,44%	57,67%	59,49
[VOC+BAH]	60,66%	62,03%	61,23
[VOC+SUN]	59,79%	57,68%	63,45

Table 5.1: Average precision of models trained with VOC2007 and with an ensemble of VOC2007 and the ETH dataset on the respective testing sets.

Subsequently, the effect of augmenting the data with motion blur is examined. Training with the VOC2007 dataset is again used as baseline. The dataset is then augmented with blurred instances of the VOC2007 dataset, creating [VOC5] for $l = 5$ pixels and [VOC10] for $l = [5, 10]$ pixels motion blur. As stated above, the motion direction θ is randomly sampled

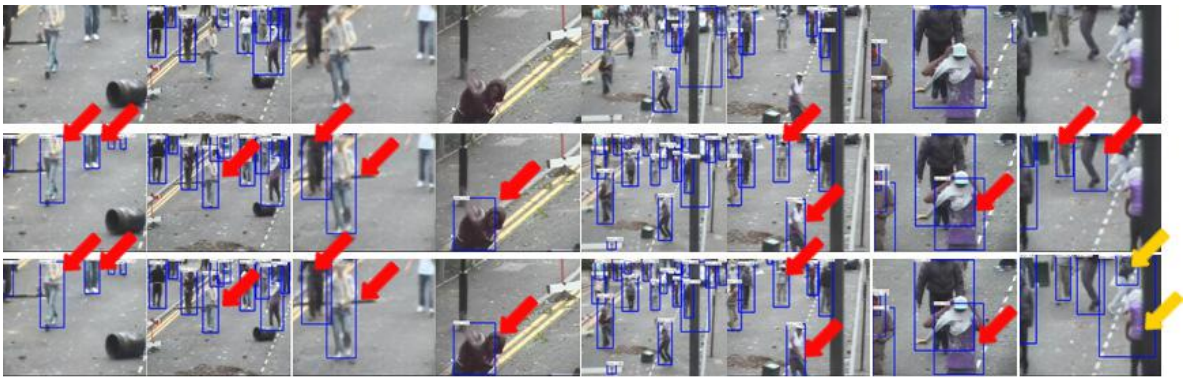


Figure 5.2: Example detections on a MET CCTV video trained with ascending levels of blurrness. VOC, VOC5 and VOC10 are depicted in rows 1, 2, 3, respectively. Red arrows depict new detections with VOC5 and yellow new detections with VOC10.

from $\Theta = [0, \pi]$. The trained models are tested on all testing sets, named $NoBlur$, $Blur5px$ and $Blur10px$ respectively. The results are depicted in Table 5.2.

	NoBlur	Blur5px	Blur10px
[VOC]	59,44%	31,71%	23,50%
[VOC5]	63,48%	62,63%	61,79%
[VOC10]	63,20%	62,31%	60,33%
[BAH]	70,70%	68,42%	59,08%
[BAH5]	70,66%	70,65%	69,23%
[BAH10]	72,52%	71,75%	71,34%

Table 5.2: Average precision of detection models fine-tuned with VOC2007 utilizing different magnitude of data augmentation on testing sets with increasing levels of blurring.

It is evident from the results that the performance of the detector is quickly deteriorating when even small amounts of motion blur are introduced. On the other hand, augmenting the training data with blurred examples is making the detector more robust, even on non-blurred data. However, when the dataset is dominated by blurred samples (VOC10) average precision declines slightly. Examples of the detection capabilities of each model on MET videos are depicted in Figure 5.2.

Figure 5.3: Example detections on a MET CCTV video. In the first row the default scaling parameter is used ($s = 600$), while in the second the detector uses a dynamically modified scaling parameter. New detections are depicted with red arrows.

Another set of experiments is performed to validate the proposed dynamic configuration of the detector. A spatial transformer layer is added in the input of the ZF network and it is applied in the region proposed by [63]. The allowed transformations (5.1) are further constrained allowing only cropping, rotation and isotropic scaling to reduce training complexity by varying s, θ in (5.6).

$$A_\theta = \begin{bmatrix} s\cos\theta & s\sin\theta & 0 \\ -s\sin\theta & s\cos\theta & 0 \end{bmatrix} \tag{5.6}$$

The transformation matrix A_t is provided to an RNN. For the RNN we use the configuration in [174]. The RNN is initially trained with artificial data created to simulate zooming operations. The set includes 100 sequences of bounding box evolution with a length of 100 frames. A linear layer is applied to convert h_t into A_{t+1}.

The transformation matrix A_{t+1} is used to predict possible severe scale changes of the objects in the next frame. The predicted scale is used to modify the scaling parameter of the detector. Initial experiments of the proposed method have been performed on the MET dataset and are depicted in Figure 5.3. The results validate the superiority of the detection performance in challenging zooming conditions.

5.2 Motion-Enhanced Object Detection

Object detection and recognition is a fundamental task for the human visual system. It has been proved that the human brain uses multiple object properties to achieve the required recognition performance. Appearance features such as shape, structure, color, and texture comprise essential information for this purpose. Therefore, most object representation methods have concentrated on single frame cues for recognition.

However, the vast majority of objects are not stationary. The motion characteristics of an object constitute a unique signature that can be used for recognition of the object. Intuitively, exploiting the motion characteristics of an object can improve our object recognition capabilities. The role of motion information in object recognition has been already examined by a number of studies [175]. Both rigid and non-rigid motion, have been studied for their role in different tasks.

As it has been highlighted, object recognition involves a number of heterogeneous modalities, namely appearance, shape and motion. Specialized neural networks have been developed to model each one of them. However, these modalities are strongly interconnected and it has been shown in the literature that employing a multi-target learning technique to address them all in parallel can have important advantages. It drastically reduces the overhead between the networks and it allows the network to generalize better.

Given the lack of motion information in single frames, only appearance-related features have been employed until now in multi-target learning methods. The shape of an object has been shown to be correlated with its motion characteristics. This is further confirmed by recent literature that has shown it is possible to predict the flow of an object *et al.*[176] from a single frame. This pseudo-flow information can be used as a substitute for the actual motion information.

In this section, a neuroscience-inspired scheme is proposed to improve object detection by introducing to the Mask R-CNN architecture an additional pseudo-temporal stream (branch) for motion prediction from still images. An object-level flow field is incorporated in the object recognition process. In particular, the proposed pseudo-temporal information is effectively incorporated into the proposed detection framework by penalizing the global loss computation with an optical flow loss factor. For this purpose, a dense pseudo-flow estimation branch is added that achieves satisfactory motion prediction accuracy at a relatively low computational cost, since the latter is applied solely at the RoI level. The proposed network detects object

(a) FlowNet **(b)** Im2Flow

Figure 5.4: Optical flow estimation architectures: a)FlowNet architecture: including the refinement part, is trained in an end-to-end manner, b) Im2Flow architecture: an encoder-decoder model that infers flow given a single image

bounding boxes with instance segmentation masks and estimates the object flow predictions for each candidate object. Appearance, shape and motion are strongly interconnected modalities and employing a multi-target learning technique to address them all in parallel can have important advantages.

5.2.1 Object-based motion analysis

CNNs have been extensively employed for optical flow estimation, achieving a huge improvement in prediction quality. In this work, the literature approach of [177] is selected (Figure 5.4a), where the information included in a pair of successive images is first spatially compressed in a contractive part of the CNN and then refined in an expanding part. However, for small displacements, *FlowNet* is not reliable. Thus, the authors proposed an extension of their previous model, called *FlowNetSD* [12], where they replaced several network parameters including kernel size and window stride of selected layers. Despite the very good results of these methods, they pose an impermeable constraint, as they require a pair of images as input to obtain satisfactory results. On the contrary, inspired by the aforementioned neuroscientific notion of visual dynamics, Gao *et al.*have introduced an encoder-decoder CNN (refer to Figure 5.4b) equipped with a novel optical flow encoding scheme that is able to translate a single static image into an accurate flow field. Their main idea is to learn a motion prior over short-term dynamics from a large set of videos and transfer the learned motion from videos to static images to infer their motion. The current study adopts the findings of Gao *et al.*[176] for object-level flow estimation.

Figure 5.5: An example of a computed flow field given a static image

5.2.2 Motion flow for object detection

The baseline of this work is Mask R-CNN that belongs, as briefly stated in section 2.3, to the Region-based/two-stage approaches. The latter is equipped with an RPN mechanism in the first stage in order to propose candidate RoIs. In the second stage, another part of the network takes the proposed RoIs and locates the relevant areas of the feature map by utilizing a RoIAlign layer. The extracted features are further processed in parallel to perform classification, bounding box regression and instance-level semantic segmentation. Both stages are connected to the backbone network.

The proposed approach mimics the visual perception procedures that take place in the human brain, following an appropriate deep neuro-physiologically grounded architecture. The primary visual cortex is emulated by the backbone of the network, generating high-level feature representations, while the dorsal ("where") and ventral ("what") stream are incarnated by the flow-estimation and object classification branches respectively, predicting object categories with each respective motion in a collaborative way.

The introduced Flow R-CNN exhibits the following advantageous characteristics: a) it enhances the two-stage detector by introducing an additional pseudo-temporal stream, and b) it incorporates the aforementioned stream in a multi-task learning process. In particular, the current study adopts the findings of Gao *et al.*[176] while moving their concept one step further, utilizing the pseudo-temporal object-level motion patterns combined with the appearance/contextual information to distinguish objects in still images. To this end, an object detection architecture is designed that takes into account the implied movement estimation.

The proposed Flow R-CNN model is built upon the Mask R-CNN model, estimating a per RoI flow field given a single static in an Im2Flow inspired branch. As a feature extractor

(backbone) the ResNet variant is selected. The existing branches of the baseline model (object classification, bounding box prediction, and mask prediction) remain intact and a 4th sub-network is integrated to the RoI head, in an end-to-end manner, to estimate a flow field for each predicted region proposal. The flow branch is inspired by the encoder-decoder logic of the Im2Flow model, where a motion prior learned from videos utilizing several urban scene understanding datasets is transferred to images, bridging the still-image detection with video object motion understanding.

Prior to the application of the proposed Flow R-CNN, the optical flow is estimated for the training datasets. More specifically, videos from several datasets were used to model the motion patterns of objects and people in the scene, and the resulting knowledge was embedded into a 2D representation for each individual image (as depicted in Figure 5.5). Finally, the proposed Flow R-CNN combines the appearance information from the static image and the predicted motion dynamics from the newly introduced branch in order to improve the detection accuracy. A graphical representation of the developed Flow R-CNN model is illustrated in Figure 5.6.

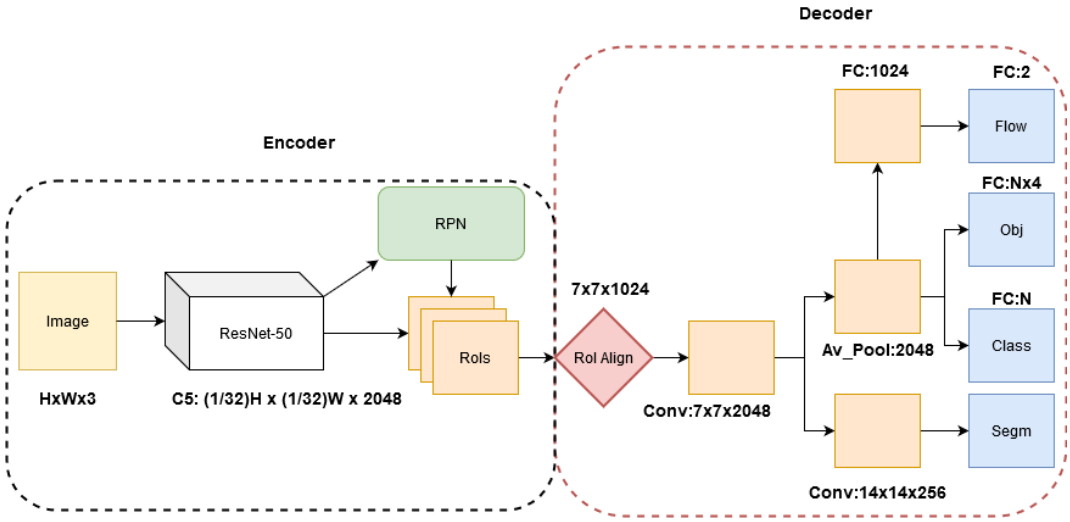


Figure 5.6: Overall Flow R-CNN architecture: a composite region-based object detection model, the backbone of the network is used for the image decoding, while the object-level flow estimation branch is used to infer the optical flow field. Sketched part of the network, i.e. the segmentation branch, remains intact during training.

It needs to be highlighted that in the current implementation a modified version of the Im2Flow was used, where the encoder part was replaced by a region-based model backbone (ResNet) and the decoder part with the object-level flow estimation branch. The developed CNN branch consists of one convolutional layer, which models the correlations among the RoI features, an average pooling layer, and two fully connected layers, for computing the respective flow field. Ablation with additional configurations, regarding the number of convolutional and fully connected layers, did not lead to improved recognition performance.

In the training phase, a multi-task loss L_{total} is defined on each sampled proposal, as shown in (5.7). The classification loss L_{class}, the bounding-box loss L_{bbox} and the instance segmentation loss L_{seg} are identical to the ones define in Mask R-CNN model.

$$L_{total} = L_{class} + L_{bbox} + L_{seg} + L_{of} \tag{5.7}$$

For each RoI, an additional object-level flow loss L_{of} is computed to supervise the per-object motion by penalizing the predicted optical flow output. This requires optical flow data for every image in the database. In datasets where optical flow information was not available, it has been estimated using state-of-the-art methodologies [12]. $L1$ between the estimated object-level flow and the ground truth one was used as a loss function.

It is argued that the loss of the optical flow estimation branch enhances the learning process of the composite model while retaining key parts of the baseline model unaffected. Experimental evaluation of the proposed method is provided in the next section.

5.2.3 Experimental Evaluation

5.2.3.1 Evaluation framework

Major research efforts have been made in the field of computer vision to understand the complex urban scenarios. The respective progress is bonded with the availability of vast amounts of annotated training data (*e.g.*cars, bicycles, pedestrians *etc.*) under varying conditions. In this section, experimental results, as well as comparative evaluation from the application of the proposed object detection method, are presented. For the evaluation, the following datasets were used:

KITTI dataset [178]: The KITTI object detection benchmark consists of 7481 training images and 7518 test images, comprising a total of 80.256 labeled objects. All images are color and the goal of the challenge is to detect objects from three common urban categories, namely *Car*, *Pedestrian*, and *Cyclist*. For evaluation, an Average Precision (AP) is computed.

V-KITTI dataset [179]: The Virtual KITTI dataset contains 50 photo-realistic high-resolution synthetic videos for a total of approximately 21.000 frames, generated from 5 different virtual words in urban settings under different imaging and weather condition. These worlds were created using the Unity game engine and a novel real-to-virtual cloning method. These photo-realistic synthetic videos are automatically, exactly, and fully annotated for 2D and 3D multi-object tracking and at the pixel level with category, instance, flow, and depth labels. For the particular task of object detection the V-KITTI contains detailed class annotation for the objects of interest (*Car*, *Van*).

Visdrone dataset [180]: The Visdrone benchmark dataset consists of 288 video clips formed by 261.908 frames and 10.209 static images, captured by various drone-mounted cameras, covering a wide range of aspects including location, environment (urban and country), objects, and density (sparse and crowded scenes). The dataset was collected using various drone platforms, in different scenarios, and under various weather and lighting conditions. From those only 8.559 images are used for the object detection task, with more than 540k bounding boxes in ten predefined categories, such as such as *Pedestrians*, *Cars*, *Bicycles*, and *Tricycles*. The dataset is further divided into training, validation and testing sets, having 6.471, 548 and 1580 images, respectively.

Cityscapes dataset [181]: The Cityscapes dataset contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high-quality pixel-level annotations of 5.000 frames in addition to a larger set of 20.000 weakly annotated frames. A number of 30 visual classes for annotation were defined, which are further grouped into eight categories: flat, construction, nature, vehicle, sky, object, human, and void. However, instance-level labeling is available only for humans and vehicles (*Person*, *Rider*, *Car*, *Truck*, *Bus*, *Train*, *Motorcycle*, and *Bicycle*). Around 3000 images are used for the training, 500 for the validation, as well as 1500 images with annotation being held for benchmarking purposes.

Berkeley Deep Drive (BDD) dataset [182]: The BDD dataset is a new driving dataset comprised of over 100K videos with diverse kinds of annotations including image-level tagging, object bounding boxes, drivable areas, lane markings, and full-frame instance segmentation. The dataset possesses geographic, environmental, and weather diversity, which is useful for training models so that they are less likely to be surprised by new conditions. The latter

contains 10 object categories (*bus, traffic light, traffic sign, person, bike, truck, motor, car, train*, and *rider*) spread over 100.000 images with over 1.8M object instance labeled bounding boxes, making it suitable for robust object detection and semantic instance segmentation. The dataset is divided further into 3 domains, namely "clear weather", "city street" and "daytime". The current study selects only the "city street" as a training domain which has a number of around 36.000 images in the training set.

Udacity dataset [183]: The UDacity dataset contains over 600K urban objects in a variety of outdoor urban videos involving *Pedestrians, Cars, Bicycles* and other objects moving in the scene. Part of the data was collected using an HD camera mounted in a vehicle. Around 375.000 annotated objects for 100k images are used for training. The train/validation and test splits are 40%, 40% and 20%, respectively.

Extensive experiments and thorough comparative evaluation provide detailed insights to the problem at hand and demonstrate the added value of the involved object-level motion branch. The overall proposed approach achieves improved performance in the six currently broadest and most challenging publicly available semantic urban scene understanding datasets, surpassing the Mask R-CNN baseline method.

A common experimental protocol was followed. Images were resized such that their scale (longer edge) is 512 pixels. The RPN anchors span 5 scales and 3 aspect ratios, and the IoU threshold of positive and negative anchors was 0.7 and 0.3 respectively. The backbone was pre-trained using the COCO [125] dataset, while for the fine-tuning, the training and validation sets from each dataset were used. As in Mask R-CNN, a RoI was considered positive if it has Intersection over union (IoU) with a ground-truth bounding box of at least 0.5, otherwise it was discarded as negative. The optical flow loss L_{of} was defined only on positive RoIs. During training, a set of 64 samples was selected for each input image, while at test time the proposal number was set 300 followed by an NMS mechanism. The NMS process was performed twice, at the RPN results as well as at the predicted classes (class-specific NMS). The optical flow branch was then applied to the top 100 detection boxes. The training phase is divided into two stages: a) only the flow branch being trained, b) all layers from ResNet stage 4 and up being fine-tuned. The model was trained using SGD, utilizing batches of 2 images with learning rate (lr), initially set equal to $1e^{-3}$. Momentum was set to 0.9 and weight decay to 0.0001. The Keras 2 framework with Tensorflow backend was used for experimentation on two Nvidia GeForce GTX TITAN X GPUs. Ubuntu 18.04.

In Tables 5.3, 5.4, 5.5, 5.6, 5.7 and 5.8, quantitative object detection results are given in the form of the mean Average Precision (mAP). The current study follows the evaluation protocol

Figure 5.7: Object detection results obtained from the application of the Mask R-CNN (upper) and Flow R-CNN (lower) models to the supported datasets.

defined by COCO challenge and adopts the primary challenge metric mAP that computes mAP over all classes and over 10 IoU thresholds. Averaging over the 10 IoU thresholds rather than only considering a single threshold of $mAP^{IoU=.5}$ tends to reward models that are better at precise localization.

For better insight, indicative object detection results obtained by the application of the proposed approach against the baseline are presented in Figure 5.7. It can be observed that the proposed scheme exhibits improved recognition performance (especially in the case of moving cars) over the baseline in various urban scenarios (night-view, top-view, car-view).

5.2.3.2 Performance evaluation

In this section, the results on all datasets are provided and discussed. The results from the KITTI dataset are presented first in Table 5.3. It can be seen that the proposed Flow R-CNN model slightly improved the results of the respective Mask R-CNN model in all categories ("Car", "Pedestrian", "Cyclist") as well as in every application scenario (easy, moderate, hard). More specifically, there was a significant improvement for the category "Car" , about 1.2%, that strongly supports the initial claim given that cars have a well defined motion pattern that can be distinctive for recognition purposes. The same applies for the V-KITTI experiments (Table 5.4), where the proposed architecture surpasses the baseline by more than

2%.

Table 5.3: Comparative results on KITTI dataset

	Easy		Moderate		Hard	
	Mask	Flow	Mask	Flow	Mask	Flow
Car	0.893	0.905	0.843	0.849	0.733	0.736
Pedestrian	0.804	0.812	0.672	0.677	0.619	0.622
Cyclist	0.739	0.746	0.635	0.638	0.554	0.556
mAP	0.812	0.821	0.717	0.721	0.635	0.638

Table 5.4: Comparative results on V-KITTI dataset

Class	Mask R-CNN	Flow R-CNN
Car	0.932	0.958
Van	0.917	0.940
mAP	0.924	0.949

Table 5.5: Comparative results on Visdrone dataset

Class	Mask R-CNN	Flow R-CNN
Pedestrian	0.205	0.223
People	0.071	0.064
Bicycle	0.029	0.033
Car	0.406	0.428
Van	0.208	0.232
Truck	0.148	0.181
Tricycle	0.132	0.148
Awn	0.091	0.085
Bus	0.216	0.253
Motor	0.153	0.151
mAP	0.166	0.180

Table 5.6: Comparative results on Cityscapes dataset

Class	Mask R-CNN	Flow R-CNN
Person	0.345	0.364
Rider	0.271	0.307
Car	0.488	0.505
Truck	0.296	0.306
Bus	0.401	0.387
Train	0.302	0.252
Motorcycle	0.237	0.256
Bicycle	0.182	0.204
mAP	0.315	0.323

Concerning the Visdrone experiments (Table 5.5), it can be observed that the introduced scheme perform reasonably well in categories where the motion is evident ("Car", "Van", "Track", *etc.*), while fails to recognize those that have complex structure or cover small portion of the image due to camera positioning (*e.g.* based on drone footage).

The exhibited results of the Cityscapes dataset (Table 5.6) suggest that incorporating the flow stream into the learning process of an R-CNN architecture may have a positive impact in the detection and recognition of moving objects, such as "Cars", "Motorcycles" and "Trucks", by 1.7%, 1.9% and 1%, respectively. Moreover, regarding the BDD experiments (Table 5.7) the influence of the motion branch is evident in the presented results, as most classes have superior recognition performance, whereas a slight increase is reported for the overall mAP (0.3%), as static objects ("traffic-light", "traffic-sign") over-shade the performance.

The last set of experiments (Table 5.8) has quite similar content and category types to the previous one, and due to the lack of optical flow training data for this group, it was decided to transfer the acquired knowledge. This limitation has led the model to fail in most cases, except in the case of cars, that hold a significant portion of the dataset, demonstrating the need for data but also highlighting the cumulative capabilities that the introduced model offers to the moving objects.

An evaluation of the proposed Flow R-CNN in six different datasets using different backbones is shown in Table 5.9. It can be observed that the proposed model achieves improved performance using deeper ResNet architectures, while benefiting from advanced schemes such as the FPN-variant, highlighting the generalization of the proposed design.

Table 5.7: Comparative results on BDD dataset

Class	Mask R-CNN	Flow R-CNN
Bike	0.383	0.391
Bus	0.481	0.489
Car	0.732	0.746
Motor	0.194	0.198
Person	0.531	0.537
Rider	0.349	0.352
Traffic-light	0.479	0.473
Traffic-sign	0.558	0.547
Truck	0.506	0.514
mAP	0.421	0.424

Table 5.8: Comparative results on Udacity dataset

Class	Mask R-CNN	Flow R-CNN
Bike	0.625	0.629
Bus	0.949	0.951
Car	0.724	0.736
Motorbike	0.738	0.736
Person	0.747	0.752
Traffic-light	0.502	0.498
Traffic-sign	0.701	0.696
mAP	0.712	0.714

Table 5.9: Comparative results on six datasets using different backbone architectures

Backbone	KITTI	V-KITTI	Visdrone	Cityscapes	BDD	Udacity
ResNet-50	0.724	0.949	0.180	0.323	0.424	0.714
ResNet-101	0.731	0.956	0.185	0.329	0.430	0.720
ResNet-50-FPN	0.735	0.961	0.194	0.334	0.432	0.725
ResNet-101-FPN	0.742	0.967	0.207	0.340	0.438	0.731

Concluding this chapter, the performance and robustness of the object recognition task on surveillance video footage has been studied. Novel methods have been proposed to improve current SoA methods in terms of speed, by dynamically adjusting the object proposal mechanism, and robustness, by exploring data augmentation techniques to alleviate motion blur introduced by PTZ operations. Moreover, a novel approach that incorporates the motion profile of an object to classify it. The proposed methods are described in detail and validated in publicly available datasets. In the next chapter, tracking of the detected objects will be studied.

Chapter 6

Multi-Object Tracking in Surveillance Footage

After identifying the objects present in the scene in Chapter 5, tracking is the next building block towards analytics for surveillance applications. Trajectories are essential to identify critical events. A literature review of object recognition methods is provided in Section 2.4.

This chapter presents the main contributions to the research community towards improving multi-object tracking. A modular multi-modal framework to alleviate trajectory fragmentation is proposed that can encode multiple characteristics of an object, including their evolution in time. The proposed method is described in detail and validated in publicly available datasets.

6.1 Introduction

The task of Multi-Object Tracking (MOT) consists of detecting multiple objects at each time frame and matching their identities in subsequent frames, yielding a set of trajectories over time. Multi-object tracking, however, has to face a series of challenges. Some of the toughest challenges are related to occlusions during which the target object is covered by outliers for an uncertain period of time. This causes trajectory fragmentation because when the target

reappears, a new identity is assigned to it. The problem is getting more intense in cases of long term occlusion because the object can re-appear at a position very distant point from the previous one. Failure to distinct occluded and occluders can lead to significant loss in tracking precision. Trajectory fragmentation can be also caused by changes in target appearance (scale, illumination, pose etc.). Keeping a unique track id for each object instance is challenging. An identity switch can be caused when objects with similar appearance concur or abrupt velocity changes occur.

We follow the "tracking-by-detection" paradigm where detections from consecutive frames have to be temporally associated. As input, the detections are produced by an object detector. However, in videos with crowded scenes, occlusions, noisy detections (e.g., false alarms, missing detections, non-accurate bounding boxes that contain multiple object instances due to localization errors), and appearance variability are very common. We follow a modular approach to achieve object tracking, combining speed and performance. Given a new frame, the tracker computes the similarity scores between the already tracked targets and the newly detected objects. A fast approach is to perform detection association using the intersection over union between bounding boxes of consecutive frames extracting small trajectories that have a high degree of confidence, a.k.a. tracklets.

Complete trajectories can, then, be formulated by matching tracklets with a more sophisticated approach that uses multiple modalities and Recurrent Neural Networks to reason on their association. More specifically, LSTM networks are used to model targets using their appearance, their volume and position in frames, their velocity and their interaction with nearby objects. Features are extracted using modality-specific neural networks trained to maximize their discriminating power. Then, they are fused to jointly reason on the tracklet association process. The similarity matrix among candidate tracklets is formulated and solved using the Hungarian algorithm [184] to find the optimal assignments.

6.2 Multi-modal Tracklet Association

In this section, we provide details on how each information modality is contributing to the tracklet association problem and how all of them are fused to define the multi-modal association solution. For this work we assume tracklets of at least 10 frames.

6.2.1 Appearance

Appearance is a dominant information cue for tracklet association. To extract appearance features, we train a Siamese network that compares pairs of objects based on their convolutional features. The neural network is trained to identify if the two tracklets, given as input, belong to the same person or not. This network takes as input images of size 100x100x3 each and estimates the probability [0,1] of these tracklets to belong to the same person (1= same, 0 =not same). The architecture of this network, inspired from [185], is depicted in Figure 6.1. The developed network learns to extract features that can model the appearance evolution of a person from different camera angles while distinguishing it from other people. Therefore, the features learned can also be used for re-identification purposes.

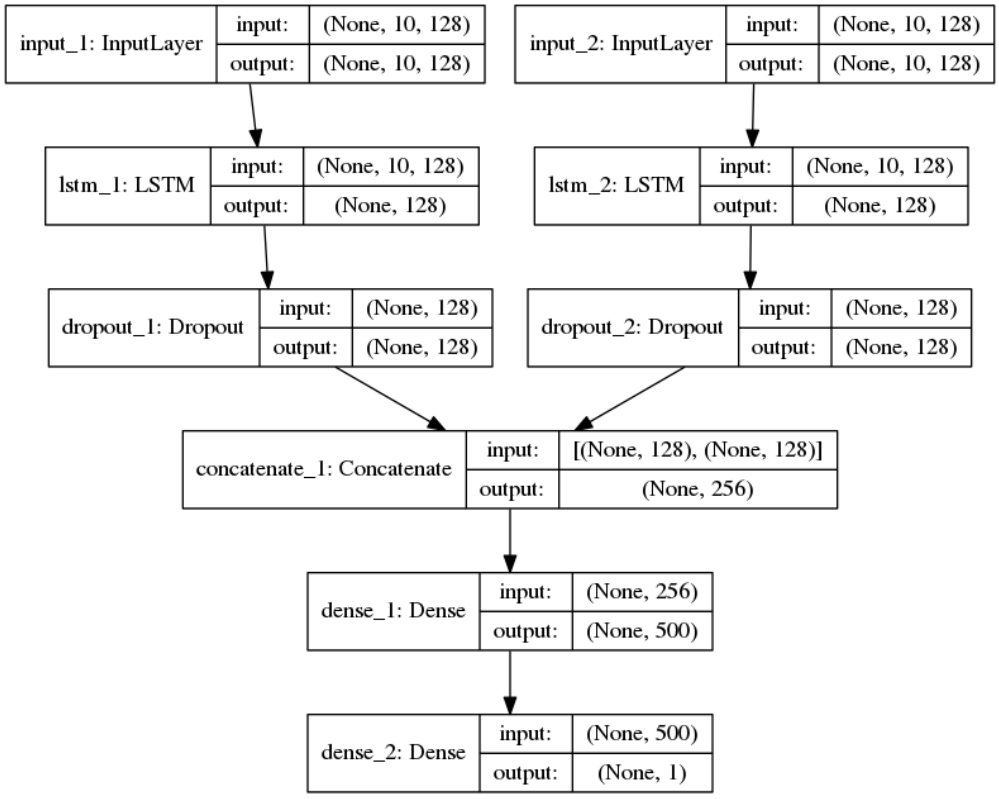


Figure 6.1: Detailed Siamese network architecture developed to model object appearance.

6.2.2 Positioning

The position of an object for the duration of the tracklet is another important cue. Features are extracted to describe the object's position sequence using a neural network. The input of network is the object's (x, y) coordinates for each frame, defined by the center of its bounding box. Note that coordinates are normalized before entering the network, i.e. x and y are divided by video width and height, respectively. A Siamese architecture, depicted in Figure 6.2 is utilized to train the feature extraction network aiming to capture the temporal evolution of the position using the LSTM layer and identify the probability of a pair of tracklets to belong to the same person.

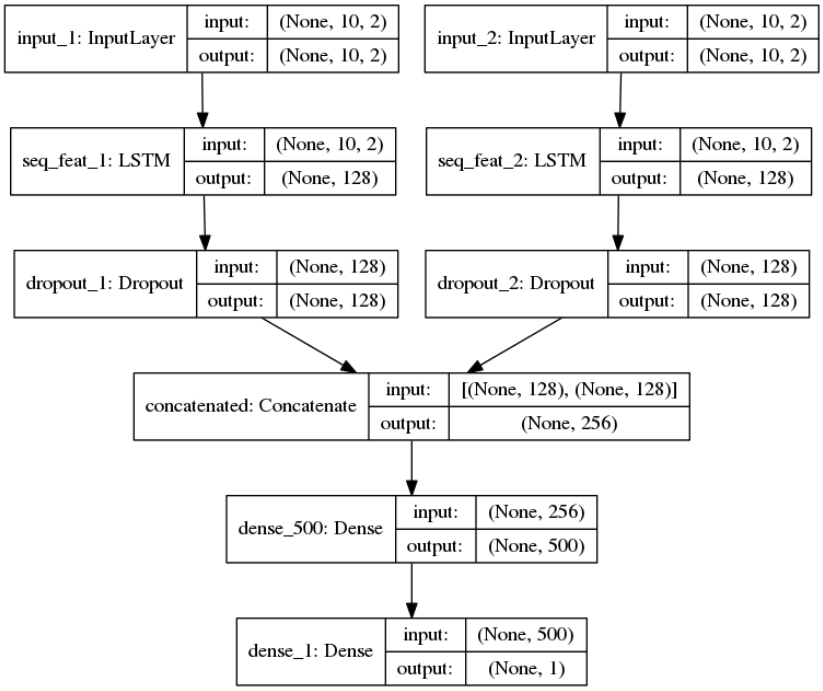


Figure 6.2: Detailed Siamese network architecture employed to model object positioning.

6.2.3 Object volume

Besides the position of an object, its volume is also important as it provides information on its distance from the camera. The evolution of the objects' volume across all the frames of the tracklet is also modeled using a neural network and a feature vector is extracted. The input of the network for each frame is the object's width and height. A Siamese architecture, depicted in Figure 6.3 is utilized to train the feature extraction network aiming to capture the temporal evolution of the volume. As in the previous cases, an LSTM layer is used to capture the temporal aspect. The network produces the probability of a pair of tracklets to belong to the same object.

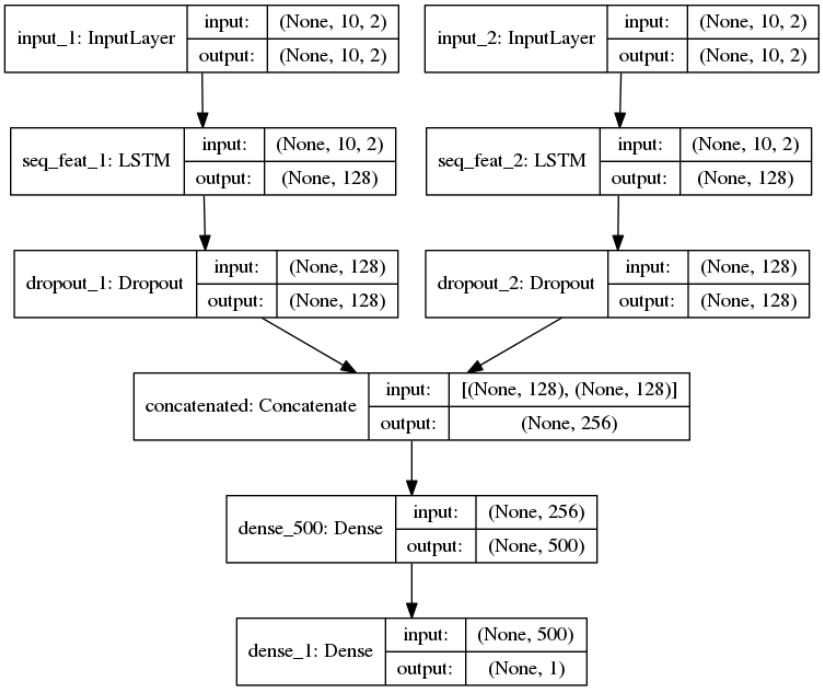


Figure 6.3: Detailed Siamese network architecture employed to model object volume.

6.2.4 Velocity

Object velocity can also be used to disambiguate the object association problem. Features are extracted to model the velocity of an object for the duration of the tracklet. The architecture of the network is depicted in Figure 6.4. The network's input is the object's velocity v for each frame is defined as:

$$v_{i,j} = (x_j - x_i, y_j - y_i) \tag{6.1}$$

where (x_i, y_i) and (x_j, y_j) are the two positions of an object in consecutive frames i and j. A Siamese architecture is utilized to train the feature extraction network aiming to capture the temporal evolution of the velocity using an LSTM layer and maximizing the discriminative capability of the network.

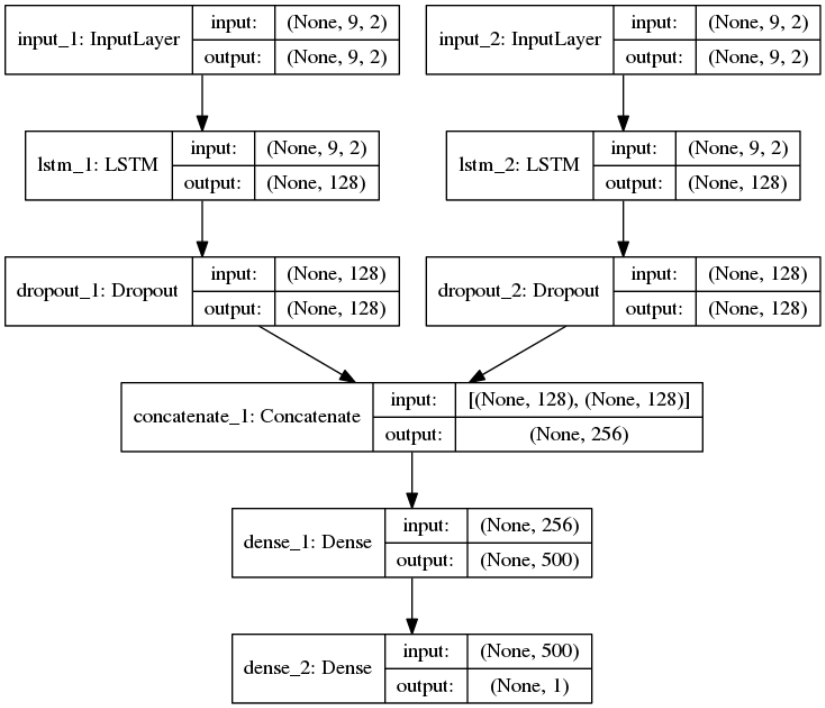


Figure 6.4: Detailed Siamese network architecture employed to model object velocity.

6.2.5 Social interaction

Finally, the social interaction of an object with others in its neighborhood can vastly improve our spatial awareness and, thus, the object association performance. For this purpose, a 15x15 image grid is created for each frame by uniformly sampling the image and a 7x7 sub-grid centred around a specific person is used as an input to the network (Figure 6.5). In [186], it is claimed that the motion of a particular target is governed not only by its own previous motion, but also by the behavior of nearby targets. Since the number of nearby targets can vary, the neighbourhood of each target is modelled as a fixed size occupancy grid. The evolution of the grid, as a model of the social interaction of the tracked object, is modeled with a neural network, similar to the other modalities. The Siamese architecture, depicted in Figure 6.7 is utilized to train the feature extraction network to estimate the compatibility of occupancy grids between different tracklets.

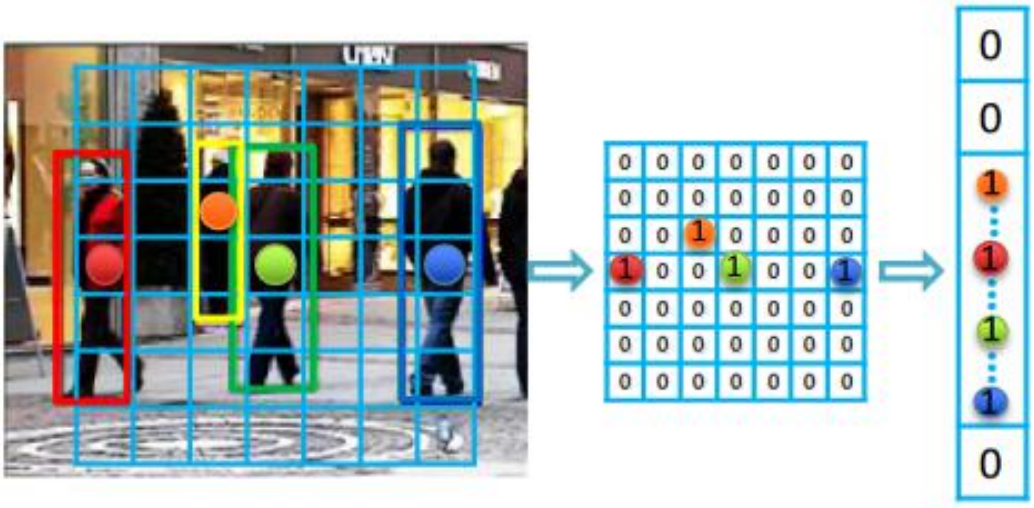


Figure 6.5: Illustration of the steps involved in computing the occupancy map. The location of the bounding box centres of nearby targets are encoded in a grid –occupancy map– centred around the target. For implementation purposes, the map is represented as a vector.

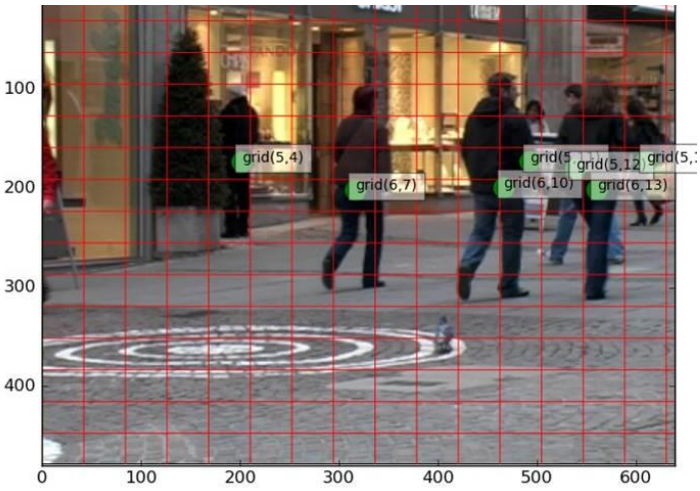


Figure 6.6: An example of the social interaction grid.

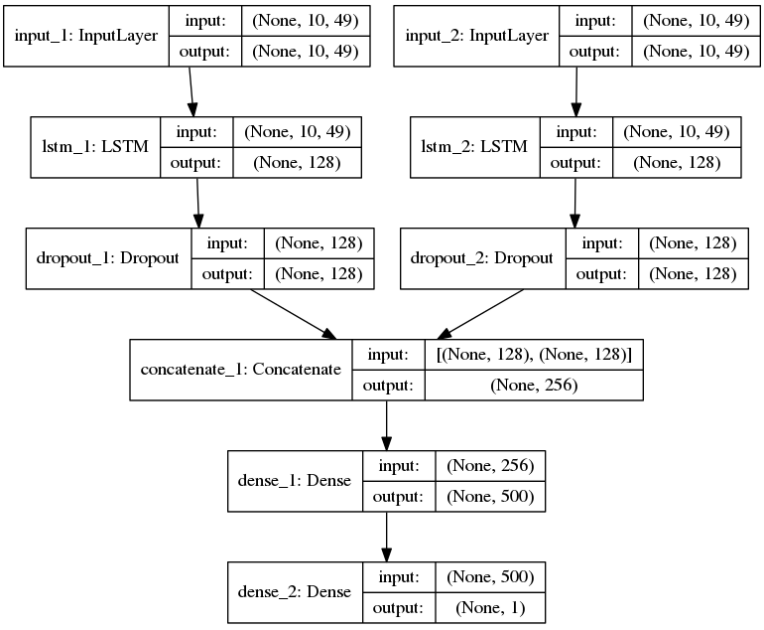


Figure 6.7: Detailed Siamese network architecture employed to model social interaction.

6.2.6 Multi-modal fusion

After modelling each modality as a separate information cue, they are all combined in a single network that fuses them providing a single decision on the matching of the candidate objects with a high prediction accuracy. The architecture of the network is shown in Figure 6.8.

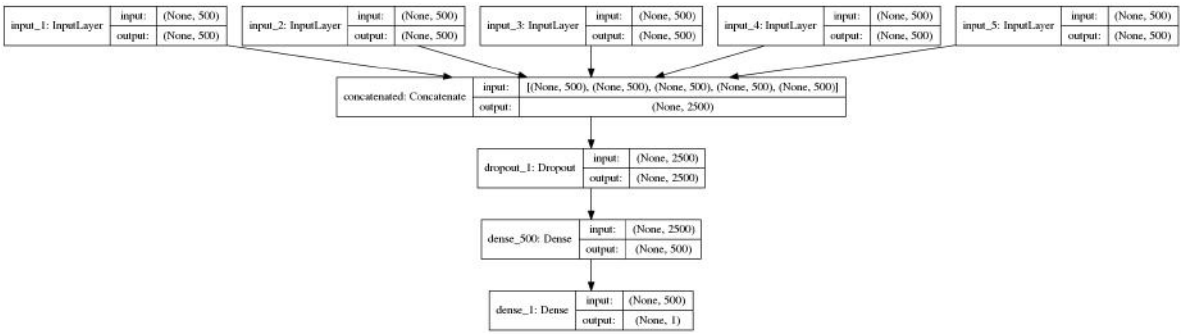


Figure 6.8: Detailed Siamese network architecture employed to perform multi-modal fusion.

6.3 Experimental Evaluation

In this section, the proposed scheme is tested and validated in a set of publicly available datasets that are closely resembling surveillance video footage.

6.3.1 Dataset

Initially, a pool of datasets was assembled by combining MOT2015, MOT2016 [187], i-LIDS [188] and PRID2011 [189] datasets to increase the size and the diversity of the samples. Especially MOT2015 and MOT2016, contain a lot of occlusions allowing the network to train on challenging cases of tracklet association. Positive and negative pairs of tracklets are created at a 1:1 analogy. Positive pairs are created from trajectories of the same person and negative from different ones but coexist in time and space. The reasoning behind this methodology is to train the re-identification network to separate hard samples having similar appearance and background, as persons appearing in the same video come close while sharing the same background. Therefore, learning to separate these samples is crucial but not trivial. From

the process described above approximately 574K pairs of persons are created. Data are split into three sets. Train (50%), Test (25%) and Validation (25%). Note that the split is carried out ensuring that a person cannot exist in more than one set to prevent over-fitting.

6.3.2 Experimental Results

In this section, the performance of each information modality on tracklet association is assessed objectively and subjectively. The fused approach combining modalities is also assessed.

Appearance: The appearance-based model is trained to successfully discriminate similar and dissimilar pairs of images with 89% accuracy. Indicative examples are provided in Figure 6.9, Figure 6.10, and Figure 6.11.

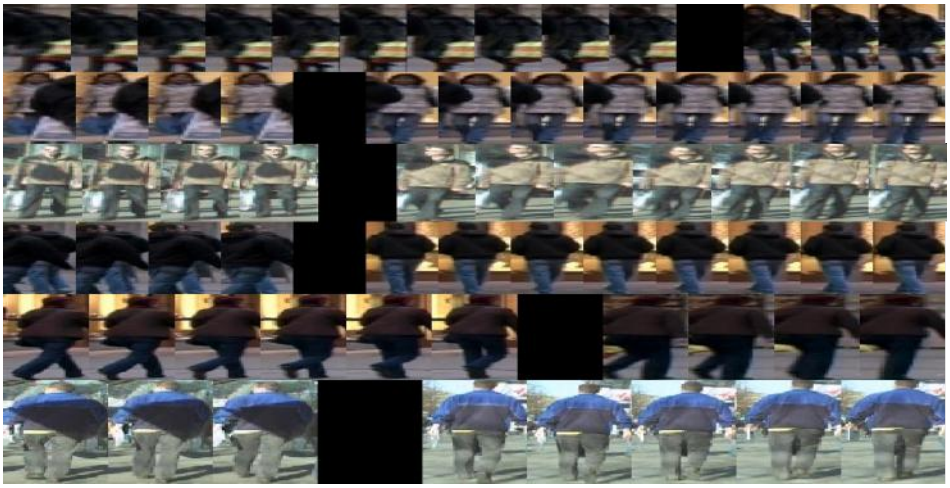


Figure 6.9: Examples of successful tracklet matching using the appearance modality.

However, in cases of heavy occlusion the appearance model fails due to the lack of information or the existence of a similar person. Another reason for failure of the model lies within the fact that in MOT2015 and MOT2016 the annotated tracks were used to create ground truth for our training set. We noticed that this may lead to the creation of dissimilar track due to the fact that the original annotation provides the position of the object even if it is completely occluded (e.g. while disappearing behind a pillar and reappearing). Given that the occluded person will not be detected by visual means, the dataset is polluted with unrealistic examples.

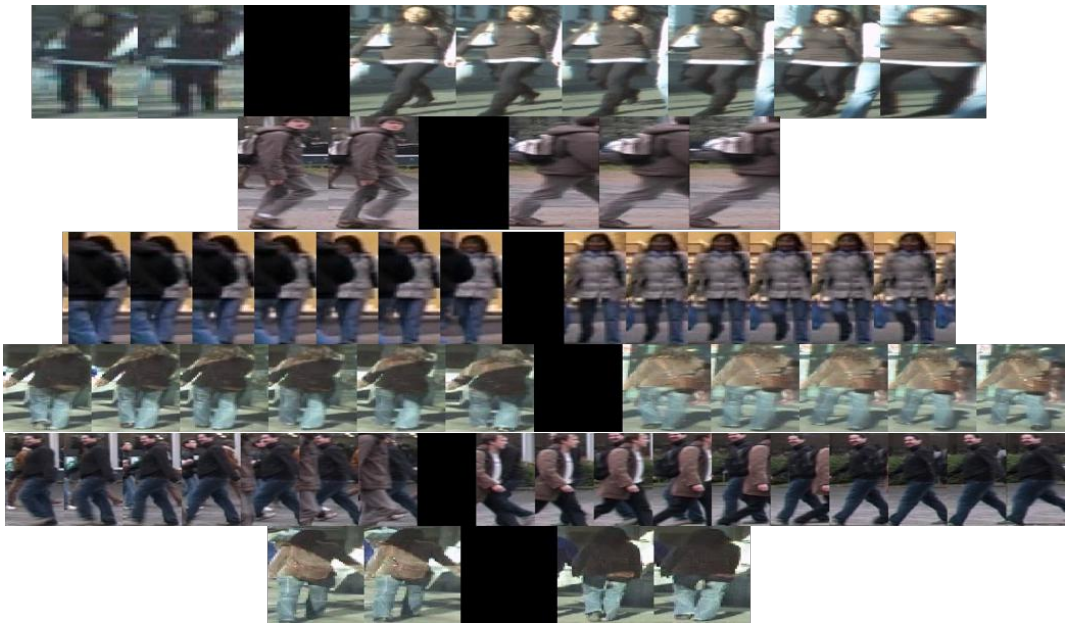


Figure 6.10: Examples of failure to match tracklets using the appearance modality.

Figure 6.11: Examples of false tracklet matching using the appearance modality.

Positioning: Tracklet association using the position information achieves 89% accuracy. The position of the tracked object in consecutive frames is one of the most important characteristics that can lead to correct predictions. Position prediction in sequences is a straightforward and effective cue that can be used for sequence prediction.

Velocity: Tracklet association using the velocity information achieves 64% accuracy, which is rather low. This can be attributed to the fact that velocity must be registered to a specific position to be meaningful as many objects can move similarly, especially in crowded environments. Moreover, camera movement immediately deteriorates the results, if not taken into account. Finally, the distance from the camera and the direction of the movement relative to it play an important role in the performance of the modality.

Social Interaction: Tracklet association using the social interaction information achieves 87% accuracy. Some indicative examples of tracklet matching are provided in Figure 6.12, 6.13, and 6.14.

Figure 6.12: Examples of correct tracklet matching using the social interaction modality. The occupied positions on each frame are depicted with white. The vertical white line in the middle separates the first sequence from the second.

Multi-modal

The model achieves 91% accuracy with a prediction speed of 26019 FPS. The combination of all cues proves to be beneficial even if some of the information cues have low accuracy on their own, such as volume and velocity, when combined with other cues they contribute to a correct final decision.

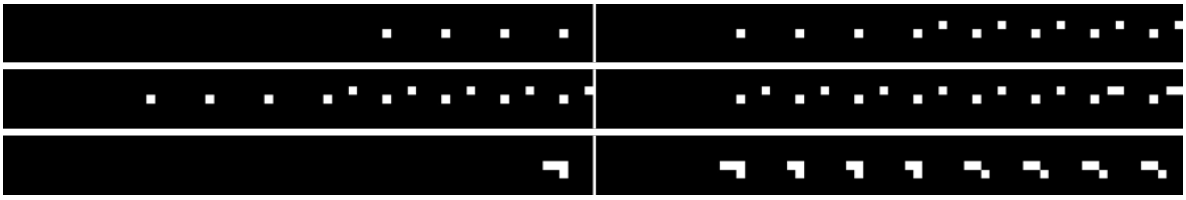


Figure 6.13: Examples of failure to match tracklets using the social interaction modality. The occupied positions on each frame are depicted with white. The vertical white line in the middle separates the first sequence from the second.

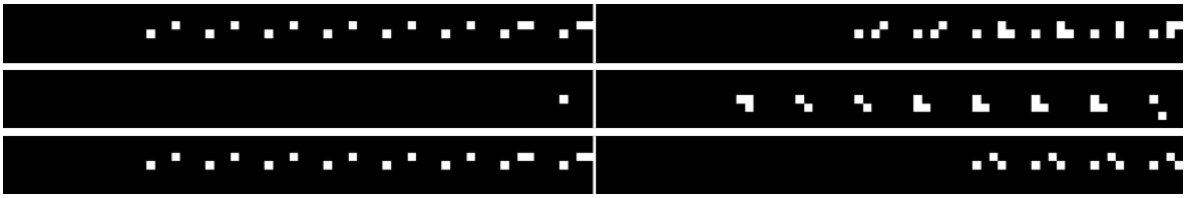


Figure 6.14: Examples of false tracklet matching using the social interaction modality. The occupied positions on each frame are depicted with white. The vertical white line in the middle separates the first sequence from the second.

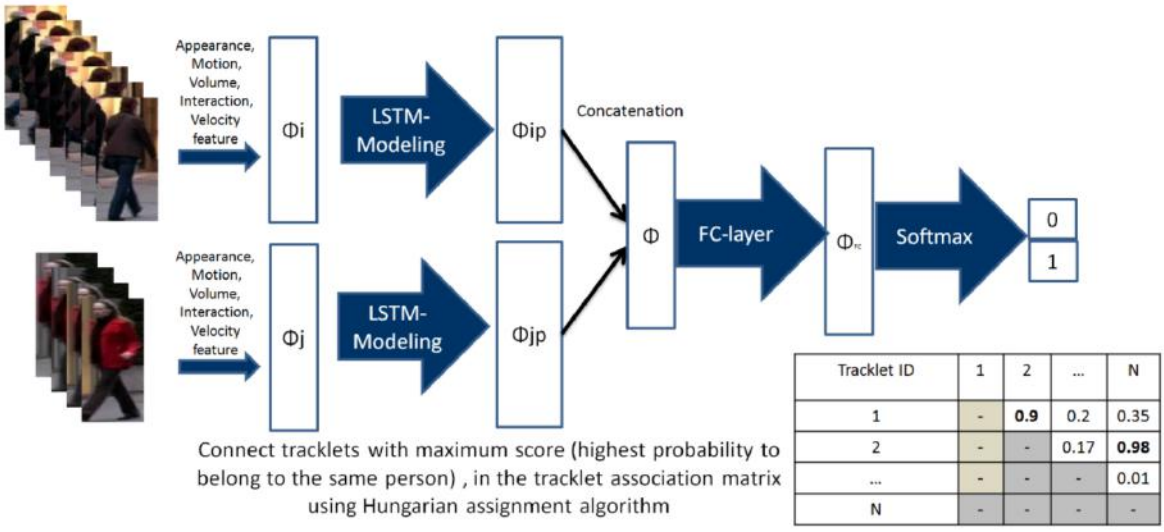


Figure 6.15: Tracklet association pipeline.

Figure 6.16: Tracklet association examples using all information modalities.

Chapter 7

Video Stabilization for Mobile Surveillance Devices

In recent years, video surveillance technology goes increasingly mobile following a wider trend. Body-worn cameras, in-car video systems, and cameras installed on public transportation vehicles are only a few cases of mobile surveillance infrastructure. Moreover, Law Enforcement Agencies are increasingly including videos recorded by mobile devices in their investigations. While, this new source of videos opens up new opportunities for the authorities, it also introduces new challenges in terms of processing, manual or automatic. Besides the huge amount of recorded footage, the produced content is usually unstable and shaken, making their manual inspection an (even more) cumbersome procedure and its automated analysis problematic due to spatial inconsistency between frames.

In this chapter, the main contributions to the research community towards stabilizing videos from mobile surveillance devices are presented. A novel method to produce stable and artifact free videos using a content-preserving warping is proposed. For this purpose, a robust optical flow is estimated by filtering out outliers caused by moving objects leveraging semantic information.

7.1 Introduction

Video stabilization is the process of generating a new compensated video sequence, where undesirable image motion is removed and has been steadily gaining importance with the increasing use of mobile camera footage. Often, videos captured with a mobile device suffer from a significant amount of unexpected image motion caused by unintentional shake of their mounting, whether this is a hand, body or vehicle. Given an unstable video, the goal of video stabilization is to synthesize a new image sequence as seen from a new stabilized camera trajectory. A stabilized video is sometimes defined as a motionless video where the camera motion is completely removed. We refer to stabilized video as a motion compensated video where only undesirable camera motion is removed. This distinction is critical since camera motion can contribute towards an aesthetically pleasing result and be instrumental for capturing the details of a scene [102].

The first step towards video stabilization involves choosing a suitable model that will adequately represent camera motion. Optical flow is the most generic motion model and recent work has shown great potential in its use for video stabilization. However, the optical flow of a generic video can be rather irregular, especially on moving objects at different depths of the scene, therefore, a motion model with strong spatio-temporal consistency and smoothness is required to stabilize the video. The approach of identifying discontinuous flows by spatio-temporal analysis and enforcing strong spatial smoothness to the optical flow neglects the semantic information of the scene contents, leading to severe artifacts when moving objects are very close to the camera or cover a large part of it. Thus, the distinction between background and foreground objects is obscured by their comparable size [30].

In this work, we are proposing the use of semantic information extracted from the examined scene together with a dense 2D motion field to produce a model representing the camera motion. The derived model allows us to generate stabilized videos with good visual quality even in challenging cases such as scenes with large foreground objects which are common in footage from mobile cameras.

7.2 Semantic filtering for video stabilization

In this work, the assumption made in [30] that the motion vector of each pixel should approximate the trajectory of the corresponding point in the scene is adopted. Given this assumption,

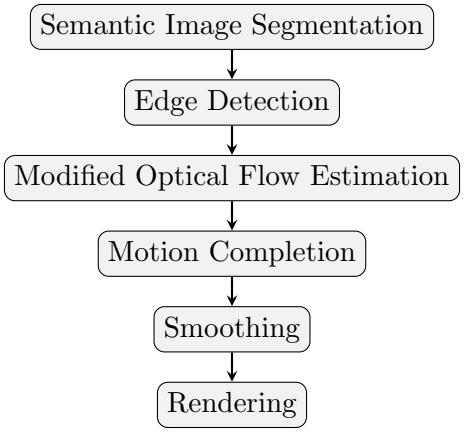


Figure 7.1: Methodology Outline

instead of smoothing feature trajectories, we can smooth the pixel profiles, where a pixel profile is defined as the accumulated optical flow vector at each pixel location. Thus, video stabilization can be achieved in a pixel-wise manner by using a pixel profile stabilization model. This assumption does not hold well, though, for scenes containing sharp depth changes and moving objects, as they can cause the optical flow field to be spatially uneven. In such cases, as it can be seen in Figure 7.2, smoothing the pixel profiles leads to artifacts. Therefore, we must modify the initial optical flow and discard the motion vectors that cause these distortions. In [30] this is performed in two iterative filtering steps, a spatial and a temporal one, trying to enforce spatio-temporal consistency.

Instead, we propose a novel method aiming to perform motion outliers rejection on the optical flow field exploiting semantic information in the context of video stabilization. For this purpose we leverage state of the art semantic segmentation [190, 191] of the scene examined to detect moving objects of interest in a surveillance scene, such as people or vehicles. Semantic segmentation masks provide the information necessary to reject irregular motion vectors, regardless of objects' size, in a single step, eliminating the need for an iterative approach and leading to a visually pleasing result.

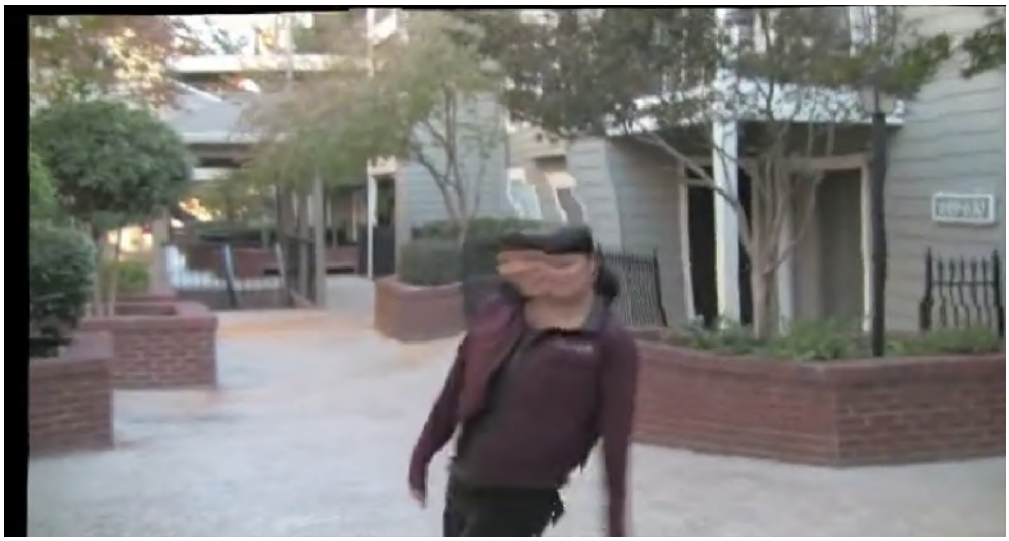


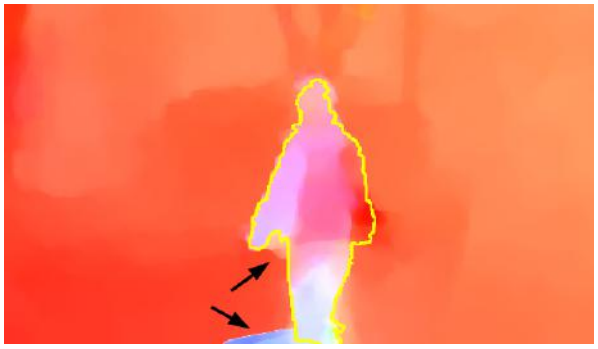
Figure 7.2: Unfiltered smoothing failure

7.2.1 Semantic optical flow refinement

Classical optical flow algorithms impose smoothness on the resulting flow field in order to solve the brightness constancy constraint equation [34]. This results in flow fields with smooth transitions between areas with different motions, producing motion irregularities within a single object. It is worth noting that this transition causes inaccurate motion vector estimation at both sides of the motion boundary, since the two motion fields influence each other. However, recent work on optical flow estimation has leveraged the use of additional information to improve flow precision, particularly at object boundaries [192, 55].

In [192] a variational energy minimization approach is employed on a grid of dense correspondences. This grid is a product of interpolation with respect to a geodesic distance, whose cost function penalizes boundary crossing. Normally, one would use an edge detection algorithm on the video frame to define these boundaries. Edge detectors that work on natural images, though, produce edges of varying strength, which do not adequately restrict the interpolation and result in flows not respecting the boundaries of our semantic segmentation (Figure 7.3b).

(a) Semantic mask



(b) Optical flow without semantic constraints

(c) Optical flow field after naive filtering

Figure 7.3: Outlier filtering without optical flow refinement. Notice in (b) the artifacts that fall out of the semantic mask, resulting in insufficiently filtered flow.

In this direction, we acquire a semantic segmentation mask for each frame in the examined video using [191] trained with the PASCAL VOC dataset that contains 21 labels including background. Given our application we are only interested in moving objects (*e.g.* "persons", "cars","motorbikes") and, thus, we discard all labels related to static objects or background (*e.g.* "potted plant", "sofa"). A naive approach would be to discard every motion vector under the semantic mask as outlier. Not surprisingly, such a method fails because of the discrepancy between the object boundaries that are delineated from the motion vectors and the corresponding ones from the semantic masks (Figure 7.3c).

Instead, we employ standard edge detection on the semantic masks, producing a set of crisp boundaries surrounding the, potentially moving, area of our frame. Leveraging the notion of geodesic distance that preserves object boundaries, we use these edges as input to the estimation of motion flow field to force the outlier vectors to reside within the boundaries of the moving object (Figure 7.4). Thus, the optical flow becomes consistent with our semantic segmentation simplifying the stabilization pipeline.

7.2.2 Motion completion

The next step is to complete the missing values of the optical flow field. We interpolate the outlier motion vectors from a grid formed in a content preserving way [106]. We use the motion vectors at the boundary of the semantic mask to form control points for the energy minimization problem:

$$E = E_d + \alpha E_s, \tag{7.1}$$

where E_d and E_s are the data and similarity terms, weighted by α. The data term is defined as a sum over all inlier points p:

$$E_d(V) = \sum_p \|V\pi_p - (p + u_p)\|, \tag{7.2}$$

with u_p being the initial optical flow at pixel p and V indicating the unknown vertices of the new grid that enclose p. π_p is the vector of bilinear coordinates of point p at the initial grid. Thus, E_d weighs toward accurate reconstruction at each data point. However, this could force the rest of the pixels to be extremely warped or distorted which is counter-weighted by the

similarity term:

$$E_s(V) = \sum_u \mid u - u_1 - sR_{90}(u_0 - u_1)\|^2, \tag{7.3}$$

$$R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

This term requires that each triangle, formed by u and two of its neighboring vertices u_0, u_1, follows a similarity transform. $s = \|u - u_1\|/\|u_0 - u_1\|$ is a term computed from the original mesh. The new vertices are calculated minimizing a standard sparse linear system. The new motion values are then bilinearly interpolated using the resulting grid.

7.2.3 Stabilization

The stable video is produced by smoothing each pixel profile independently. We do not employ an adaptation scheme for the temporal window, since all these approaches require arbitrary thresholding and are heavily influenced from the frame rate. The smoothing is achieved minimizing the following objective function:

$$O(P_t) = \sum_t \Big(\|P_t - C_t\|^2 + \lambda \sum_{r \in \Omega_t} w_{t,r} \|P_t - P_r\|^2 \Big), \tag{7.4}$$

where C is the cumulative motion vector field of the input video at frame t and P the corresponding one of the output video. $w_{t,r}$ is the weight of past and future frames r in the temporal window Ω_t and is calculated by $w_{t,r} = exp(-\|r - t\|^2/(\Omega_t/3)^2)$. The first term of this sum is the similarity between the stabilized and the initial frames, a factor that minimizes cropping, while the second term expresses the similarity of the new frame to its neighboring ones, which maximizes stability. Finally, λ acts as a balancing term that allows us to favor the one over the other.

The optimization is solved by a Jacobi-based iteration [193] for each pixel by:

$$P_t^{(\xi+1)} = \frac{1}{\gamma} \Big(C_t + \lambda \sum_{r \in \Omega_t, r \neq t} w_{t,r} P_r^{(\xi)} \Big), \tag{7.5}$$

with the scalar $\gamma = 1 + \lambda \sum_r w_{t,r}$ and ξ being the iteration index (by default, $\xi = 10$). Note that unlike Liu *et al.*our algorithm runs only once. We render the final result by warping each frame with a dense displacement field $B_t = P_t - C_t$.

(a) Original frame

(b) Optical flow with semantic constraints

(c) Proposed filtered optical flow

Figure 7.4: Outlier filtering with optical flow refinement. Notice the alignment between the motion vectors and the boundary in (b).

7.3 Experimental Evaluation

We conducted a wide range of experiments on publicly available baseline videos with moving objects, occlusions and parallax. Additionally, we experimented on videos from the surveillance domain, especially police body-cam videos, which contain highly irregular motion (*e.g.* walking, running) and occlusions, especially from persons, bystanders etc.

Our method manages to successfully filter out moving objects in the majority of cases. Figure 7.6 shows a typical failure case for most trajectory based methods, where an object covering a significant portion of the screen crosses the field of view. Naturally, such an object has a big effect on the flow field and if we stabilize the video without some way of filtering we see visible artifacts (*e.g.* the elongated head of the lady in the foreground, together with the warped body of the lady in the background in row 2). Our output is stable and without artifacts. Similarly, in the surveillance domain video of Figure 7.7, which again contains a significantly big moving object and heavy shake, one can clearly see the distortion on the face of the officer, especially on the last frame of row 2, which does not exist in our output. The presented results are qualitative, since result quantification is not a trivial matter in video stabilization, due to the fact that there are no benchmarks or widely accepted metrics available.

The experiments were run on a GTX 1070 GPU. For the initial semantic segmentation masks and optical flow we used CRF-RNN [194, 191] as well as DeepMatching [195] in conjunction with EpicFlow [192]. We empirically choose $\alpha = 1$, $\lambda = 1$ as they give the most pleasing results.

(a) Scene with many faces **(b)** Inaccurate semantic mask

Figure 7.5: Semantic segmentation failure

Figure 7.6: Typical failure case for trajectory based methods. Our system manages to stabilize this heavily occluded scene. The rows from top to bottom correspond to the original, stabilized without filtering and successfully stabilized cases. Notice the heavy distortions in the second row.

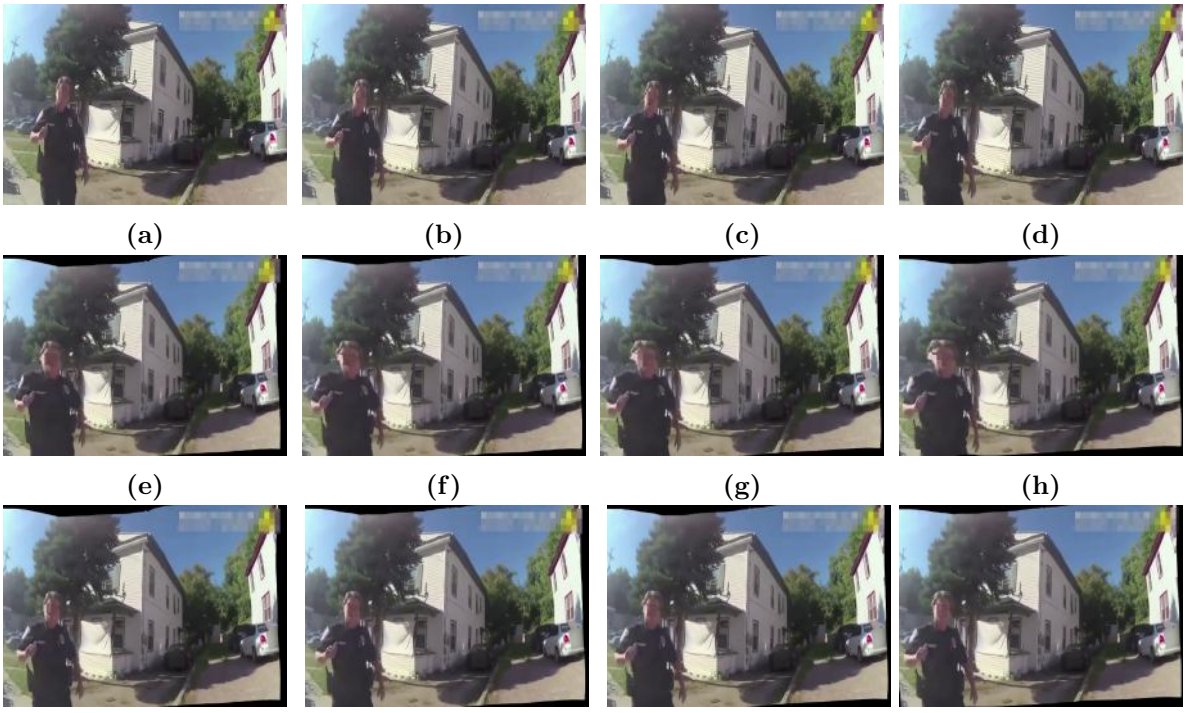


Figure 7.7: Four frames of a video in the surveillance domain. Again, the first row depicts the original, unstable, video, the second one is a stabilized without semantic filtering and the third a stabilized version with our method. Notice the distortions around the officer's head at the last row, while our results remain crisp.

Chapter 8

Real applications utilizing the proposed methods

To conclude, this chapter is intended to showcase that the proposed methodologies for object recognition and tracking are applicable in video analytics for several surveillance applications. An investigation assistant for large video archives is presented where object recognition and tracking have an integral part, enabling the semantic analysis of the investigation content to make it searchable in a user intuitive way. Moreover, the intermediate results of the methodologies proposed, such as the appearance features and the optical flow, can be used to empower new analytics. A crowd behavior analysis system is also presented that exploits the contributions of this work.

8.1 Surveillance Video Archives Investigation Assistant

The rapidly increased use of video surveillance in multiple business sectors has created new challenges related to the exploitation of surveillance video archives. Performing investigations in archives for large scale camera networks that may comprise multiple sites, complex camera topologies and diverse technologies requires significant human effort. Video analytics are destined to assist but in spite of the progress achieved, advanced analytics are still at a relatively nascent stage owing to a number of challenges; (a) large scale processing requirements, (b)

support diverse video content with differences in quality, format and extrinsic parameters, (c) inter-camera analytics, (d) effective data visualization and (e) a user-intuitive interface. The need for investigations in huge amounts of videos from multiple sources is forming a new product space for tools that can assist investigators to face the challenges in their line of work, streamlining the work of expert law enforcement officers and investigators by automating burdensome processes. Moreover, it could offer improved situational awareness and search assistance tools to further diminish the possibility of missing evidence due to the huge workload.

In the framework of the EU Horizon 2020 Fast Track to Innovation (FTI) programme (Grant Agreement No n° 720417), the SURVANT project has developed the identically named system to assist investigators to search efficiently and effectively in video archives and is expected to contribute towards fighting crime and illicit activities, improving the sense and essence of security for the citizens. SURVANT developed a product that introduces novel solutions to face the challenges identified in the markets targeted. In Figure 8.1 an outline of the system architecture is depicted. Here, we will focus on the Video Ingestor module that takes advantage of the methodologies proposed in this work.

The Video Ingestor module of SURVANT is responsible for processing videos to extract video analytics that will enable content-based retrieval functionalities to the system. The aim of this module is to detect the objects that are meaningful for LEA investigations. Furthermore, it aims to track the target objects in order to identify the route they followed before/after an event happened. The aim of tracking is to keep the same track identity for every object, from the moment that it enters a scene to the moment that it departs. Each video that is included in an investigation is automatically processed. The objects recognized and their trajectories are stored in a database, including the features extracted to encode their appearance and motion characteristics. These features are subsequently indexed, enabling content-based queries in the video repository. Due to security and privacy restrictions, it is not possible to present results from the use of the object recognition and tracking on actual content.

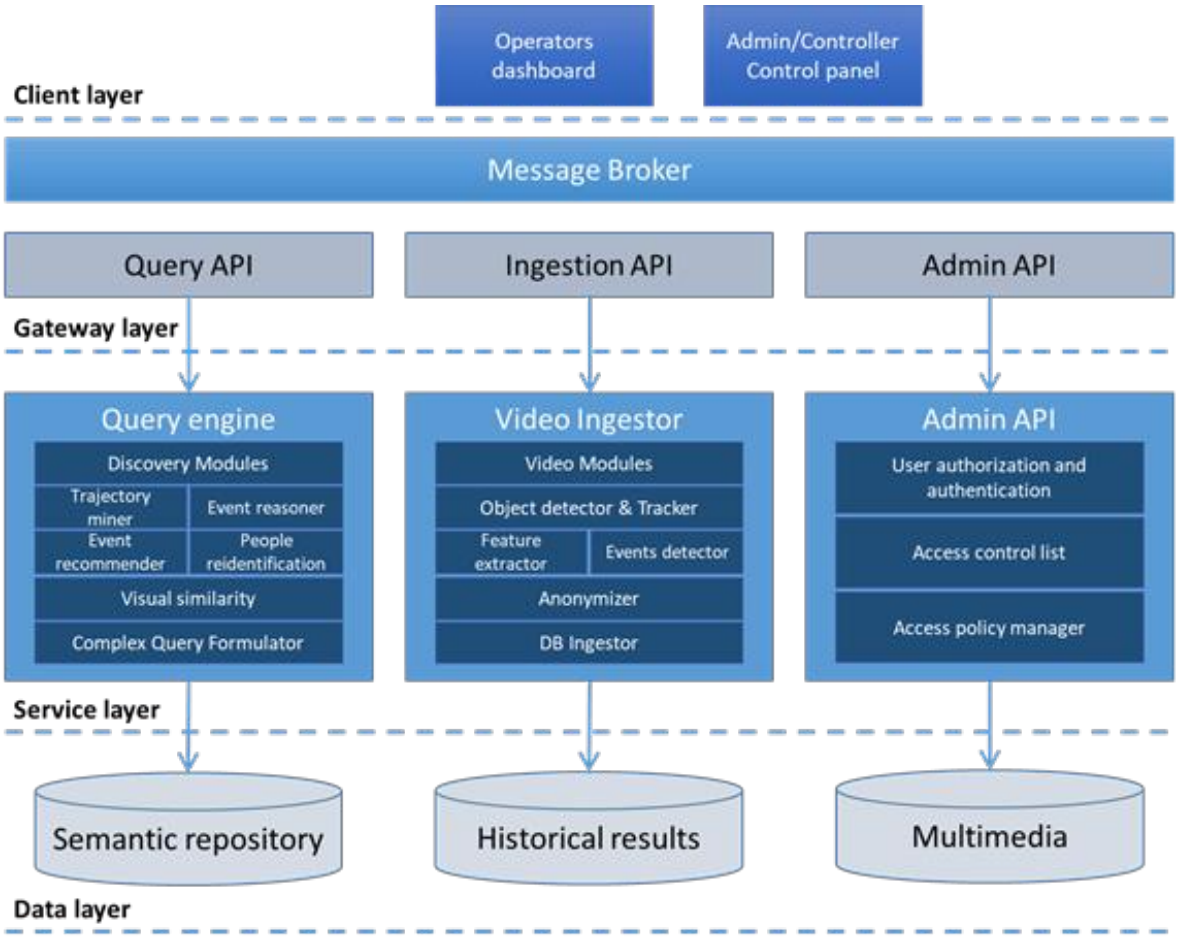


Figure 8.1: System architecture of the SURVANT investigation assistant.

8.2 Crowd Behavior Analysis

Behavior analysis is one of the most challenging tasks in computer vision. While the analysis of human activity has received a lot of attention for actions performed by individuals, work on crowded scenes has been significantly less. Crowd scene analysis faces even more challenges than individual human activity due to numerous facts. The density of people found in such scenes is often prohibitive for detection algorithms that cannot identify accurately individual entities. It is even more difficult to identify body parts and their respective motion patterns to classify the individual activity of each participant. The behavior of the crowd often exhibits emergent behaviors and self-organizing activities, especially during abnormal events. Furthermore, the available datasets are often of low quality and they are lacking real-world examples of the events to be detected as they are only available to the authorities for legal and privacy reasons.

In this section, a novel methodology for classification-based abnormal crowd behavior is presented using crowd density heat-maps as an attention mechanism for analyzing motion patterns in crowded areas of the scene. The optical flow estimation method proposed in Section 4 is employed.

8.2.1 Abnormal Event Detection

Abnormal event detection in crowded scenes is based on the analysis of the combined actions of the participating people. Due to the inapplicability of classic detection and tracking methods in highly crowded scenes, a more holistic approach is required. It is obvious that the motion content is the main source of information, either in pixel or feature level. Moreover, this motion content must be analyzed in a broader spatial and temporal context. Therefore, the optical flow of such a scene is suitable for analysis. However, such an analysis could be susceptible to errors due to similar motion content from parts of the scene where no people are present. It is argued that a crowd density heat-map can act as a driving feature to ensure that only relevant regions are included in the motion analysis. Furthermore, in the temporal domain, the changes in crowd density, such as a sudden evacuation of a place, can also provide invaluable evidence for the existence of an abnormal event.

The first step in the processing pipeline is to generate a density heat-map and an optical flow estimation. The density heat-maps are derived as described in [16] (Figure 8.3), while the

flow estimations are produced as described in Section 4. The data are input into a two-stream network depicted in Figure 8.2. The first stream takes as input the density heat-maps and the second one the optical flow compared to the previous frame. Inspired by [196], we use a convolutional spatiotemporal network to learn the regular patterns in the training videos, in time and space. In order for the network to learn the regular patterns in the training videos, we used Long Short Term Memory networks. LSTMs are employed to acquire information about spatial structures of each input frame and for learning temporal patterns of the encoded spatial structures respectively. The output of each model ends up in the respective fully connected layers and then merged. Finally, the proposed network is designed to classify the abnormal events into more fine-tuned categories (such as *Panic*, *Fight*) rather than just flagging videos for abnormal activities.

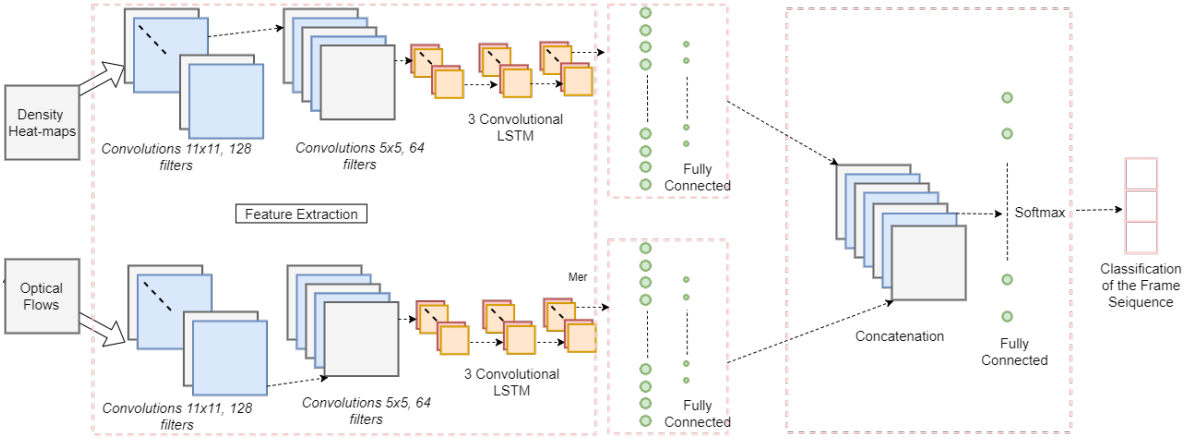


Figure 8.2: Proposed network architecture for detection of abnormal events in crowded scenes.

8.2.2 Experimental evaluation

In this section, the experimental evaluation procedure is described. The aim of those experiments is dual. First, the contribution of crowd density heat-maps to improve classification of abnormal events is examined. For this purpose, the proposed methodology is compared against previous work reported in [197] for the events of *Panic* and *Fight*. Moreover, initial experimental results are provided for the GTA dataset created in [16] using the Grand Theft

(a) Original frame from GTA V **(b)** Predicted heat-map

Figure 8.3: Example frame and the predicted density heat-map.

Auto (GTA) V game engine and Scripthook V [198], a library developed to provide access to the native functions of GTA V.

For the purposes of this work, the dataset from [197] was filtered to exclude videos with ground truth labels that are irrelevant to this work. The remaining 17 videos, have been divided in training and testing set. More specifically, we used 14 videos as training set and the rest for testing set. For each video of the training and testing set, consecutive batches of 10 frames are fed to the network. Table 8.2 illustrates the performance comparison between our approach and the evaluations which have been done from the authors of [197] in the panic and fight behavior categories. The proposed methodology is shown to significantly outperform previously reported results. More detailed results for the proposed method are presented in a confusion matrix in Table 8.1. From the confusion matrix it is derived that the proposed method achieved 87% accuracy in events of *Panic*. Compared to *Fight* events, *Panic* seems to have better results. This probably happens because *Fight* has similar macroscopic motion patterns with other common situations, such as gatherings and loitering.

We followed the same procedure for the GTA dataset. From its 14 video sequences, we chose 10 randomly for the training set and the remaining for the testing set. Again, batches of 10 consecutive frames have been derived from both of the sets. As can been seen in the confusion matrix, Table 8.3, again the panic abnormal behavior has the best results. Note that due to

Actual \ Predicted	Normal	Panic	Fight
Normal	84.8	2.7	12.3
Panic	9.6	87	3.2
Fight	56.6	0	43.3

Table 8.1: Confusion matrix of the proposed method on the Novel Violent dataset.

	Normal	Panic	Fight
DT [197]	36.8	74.8	30.4
HOT [197]	36.5	62.18	38.2
Proposed	84.8	87	43.3

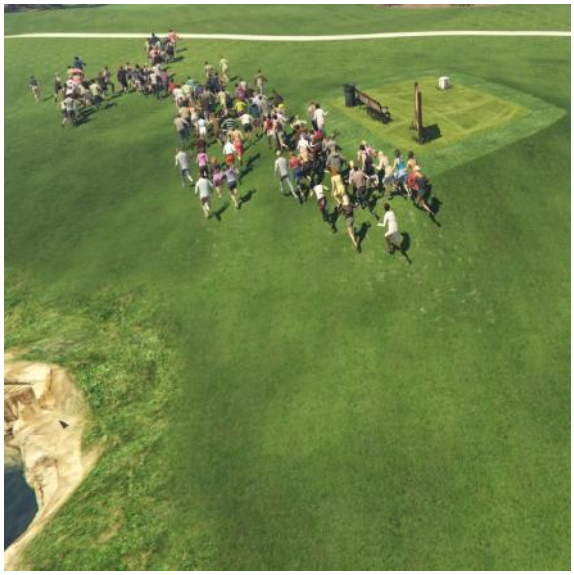
Table 8.2: Comparison of classification accuracy per class.

the density of the crowd and the variety of complex events such as celebrating and amusement, this dataset seems to achieve lower overall accuracy.

Actual \ Predicted	Normal	Panic	Fight
Normal	83.8	5.8	10.2
Panic	37.3	61.2	0.6
Fight	70.2	0.7	28.9

Table 8.3: Confusion matrix of the proposed methodology on the $GTA - Crowd$ dataset

(a) Normal Crowd



(b) Crowd in Panic

(c) Crowd in Fight

Figure 8.4: Example frames from the abnormal crowd detection dataset. For panic and fight the frames have been cropped for visibility reasons.

(a) Normal Crowd

(b) Crowd in Panic

(c) Crowd in Fight

Figure 8.5: Example frames from the Novel Violent dataset.

Chapter 9

Conclusions

This work studied research subjects related to object recognition and tracking for surveillance applications aiming to improve their performance, extend their capabilities and provide new insights. The work performed can be analyzed into some discrete steps, namely the extraction of appearance and motion features, object recognition, and object tracking. Moreover, the application of the developed methodologies to more complex video analytics applications was examined.

The first step in computer vision applications is to encode the information of the input material into features that describe it in a compact yet comprehensive way. We have studied the extraction of appearance features using CNNs and proposed a novel architecture to achieve better knowledge modeling. New insight has been provided on the use of multi-branch residual blocks to model appearance information (Section 3). The proposed CNN network has been proved to improve performance in vision applications such as object recognition, and object segmentation when used as a backbone (feature extraction network). The next most important information modality encoded in videos is the motion content of the scene. The optical flow has emerged as the representation of choice. We have studied the estimation of the optical flow and proposed a novel method to improve the quality of the output by taking into account the semantics of the examined scene. The importance of using contextual information to regularize the training of a network has been highlighted (Section 4). A consistent and detailed optical flow was shown to improve both object recognition and tracking used either as an additional information modality or as a pre-processing step to stabilize videos that suffer from camera motion (Section 7).

Diving into the heart of this work, we have studied the optimization of the object recognition process (Section 5). This was performed in two distinct ways; improving the object detection process by dynamically adjusting the object proposal mechanism, and enhancing the recognition performance by taking into account the objects' motion behavior modeled by a pseudo-temporal flow. The former one allows the object detection network to deal with the challenges introduced by PTZ operations that drastically change the size and the position of the objects in the scene. Theoretical and experimental analysis of the problem on real surveillance data has highlighted the need to utilize more flexible solutions. Moreover, the proposed solution improved the detection speed, optimizing the time-consuming object proposal process. The latter improved the performance of the object classification process as it takes into account the specific motion behavior of each object class. The importance of motion, whether this is real optical flow or a pseudo temporal flow, in object classification has been grounded. Improved object recognition performance has been demonstrated in multiple datasets.

Object tracking is also a key target of this work. More specifically, multi-object tracking was performed using the tracking-by-detection paradigm. We have studied the detection association problem and a methodology for multi-modal association was proposed (Section 6). The methodology is fusing, beyond the appearance information, also position, speed, size and interaction with other objects to reason on the association of two object instances. Moreover, it has been shown that it is beneficial to model the temporal evolution of these modalities allowing the tracker to associate even object instances that have gradually distanced in terms of appearance. A robust object tracking method is essential for analytics in surveillance applications that pose a lot of challenges such as occlusion, pose and lighting changes that drastically affect the tracking accuracy.

Finally, the proposed methodologies have been applied in real world applications and advanced video analytics tools to demonstrate their efficiency. An investigation assistance tool for surveillance video repositories is presented. Moreover, a tool for extracting crowd behvior analytics is also described. Example architectures and challenges faced have been discussed.

In this chapter, we discuss each contribution independently in Section 9.1 and its impact of the proposed work on improving video analytics capabilities for surveillance applications in Section 9.2. The thesis is concluded by discussing the limitations of the proposed work as well as new ideas and research directions in Section 9.3.

9.1 Contributions

The first contribution focused on an extension of Residual Networks to improve our capabilities to model visual information. The residual information content is transformed into a hidden state mimicking a Linear Dynamical System. This transformation is learnt based on the statistics of the training content and, therefore, it leads to a more generic representation of the information content that we claim and prove that it can be easier learnt. Therefore, the proposed LDS module essentially breaks down the learning procedure in a content-based hierarchical way. The skip connection transfers the whole image, leaving the residual modelling to the other two branches. The LDS branch is learning a generalized version of the residual information and the CNN branch is learning the residual of the residual information, which is even more sparse than the original residual information and, thus, easier to learn. An exploration of the network architecture space has been presented to assess the performance of the LDS module in multiple depths and in conjunction with other ResNet-based methodologies and extensions. The proposed LDS blocks can be utilized in any architecture that employs residual blocks, simply by replacing them. Experiments on image classification (CIFAR-10/100, ImageNet), object detection (PASCAL VOC and MOT2017Det) and segmentation (brain MRIs) datasets have demonstrated the outstanding performance and robustness of the proposed approach.

Our next contribution refers to the motion information. Novel means of regularization were explored to improve optical flow estimation neural networks in terms of generalization and ease of training. The first regularization strategy proposes the infuse of semantic information in the training process to enable a better understanding of the scene context and, subsequently, to achieve better flow estimation at inference. The proposed semantically driven local consistency regularization method is utilizing ground truth semantic information to learn to implicitly identify the semantic edges of an object and better reason on the motion vectors around them. In the context of the optical flow estimation, semantic edges proved to be a better approximation of the real "motion edges" of an object compared to edges from a traditional edge detector, as used in the literature. The proposed method is highly versatile and can be easily integrated into existing training pipelines without modifications in the inference step.

The second regularization method proposed deals with the use of pixel coordinates to regularize the training process. An implicit regularization method using the normalized coordinates of each motion vector as a feature was also proposed and experimentally assessed for optical flow estimation. This method is adding spatial awareness to the data input of the network, breaking the spatial invariance properties of the CNN. Extensive experimental evaluation

proved that this is beneficial for the training process improving speed and reducing variance, while converging to a better solution. While this regularization method requires some architectural modifications, minimal processing overhead is added during the inference stage. Finally, it was proved that both the proposed regularization methods are complementary and can be combined to further improve the results. The well established FlowNet 2 was used as a reference architecture to evaluate the added value of the proposed solution. The accuracy of the baseline flow estimation network was improved on both synthetic and real datasets, while minimally affecting the underlying architecture.

Another work presented contributes to improving multi-target object detection based on region proposals, such as the R-CNN approach. The contribution of this work is two-way; Methodologies to improve the efficiency of deep learning based multi-target detectors in challenging CCTV footage are proposed. A novel methodology to dynamically tune the detector parameters during intense PTZ operations is proposed. The affine transformation parameters of the objects, derived from a spatial transformer network, is used to train an RNN to predict the intrinsic camera properties in the next frame. The predicted parameters are used to tune the detector parameters, leading to more robust results. Initial experiments have shown that dynamic scaling significantly improves the performance of the detector compared to fixed scale operations. Moreover, the use of heterogeneous data as well as data augmentation with motion blur is explored for training object detectors. Experimental results have shown that the detector benefits from both methodologies. Robust performance was reported in both original and blurred content, as well as challenging action scenes in CCTV videos.

Continuing the contribution to object detection, a methodology for incorporating pseudo-temporal information in Region-based CNN object detection schemes was presented, in contrast to the vast majority of literature that relies only on the use of appearance information and semantic knowledge. Following a neuro-scientifically grounded scheme, the pseudo-temporal stream was integrated parallel to the classification, bounding box regression and segmentation mask prediction branches of Mask R-CNN, and it was effectively incorporated into the learning process by penalizing the global loss computation with an optical flow loss factor. Extensive experiments and thorough comparative evaluation were reported, which provide a detailed analysis of the problem at hand and demonstrate the added value of the involved instance-level motion branch. The overall proposed approach achieved improved performance in the six currently broadest and most challenging publicly available semantic urban scene understanding datasets, surpassing the baseline method.

Towards multi-object tracking, a hierarchical object association method is proposed to balance

between speed and performance. A set of tracklets featuring robust consecutive detections of objects are created using a simple IoU based method. Subsequently, a tracklet association scheme that captures the temporal evolution of five information modalities (appearance, position, velocity, volume, social interaction) is proposed. Feature extractors are trained separately for each modality using a Siamese network architecture and later combined into a unified fusion network. The fusion of all cues proves to be successful with 91% accuracy. All information modalities are contributing to the final decision, showing good complementarity.

Another contribution is focused on video stabilization and the challenges posed by moving objects in an unstable scene. A novel video stabilization pipeline was presented that leverages the latest advances in semantic image segmentation and fuses it with a dense 2D motion field to produce a model representing the camera motion and refine the calculation of optical flow. An efficient filtering technique was used to remove motion vector outliers, caused by moving objects, by leveraging semantic information. We calculate the optical flow using a content-preserving warping and a pixel-level smoothing scheme, forcing the outliers to reside in the edges of our semantic mask. This way we manage to produce stable, artifact-free videos in scenes with moving objects, occlusions and parallax. The derived model allows us to generate stabilized videos with good visual quality even in challenging cases such as scenes with large foreground objects which are common in footage from mobile cameras.

Finally, a contribution towards crowd behavior analysis was presented, aiming to develop a new method that can precisely detect and classify abnormal behavior in dense crowds. A two-stream network was proposed that uses crowd density heat-maps and optical flow information to classify abnormal events, such as *Panic* and *Fight*. Work on this network has highlighted the lack of large scale relevant datasets due to the fact that dealing and annotating such kind of data is a highly time consuming and demanding task. Therefore, a new large scale synthetic dataset has been created using the Grand Theft Auto (GTA) V engine which offers highly detailed simulation crowd abnormal behaviors. This dataset is available to the community. The proposed network has produced improved crowd behavior accuracy compared to the literature.

9.2 Impact

The present work has been evaluated by the international research community, reported and published in the form of peer-reviewed papers and in scientific journals and in the proceedings

of scientific conferences. More specifically, 3 publications have been published in Q1 journals (IEEE CSVT, MDPI Sensors, Springer Neuroinformatics) and 4 publications in IEEE conferences (ICIP, ICPR, AVSS, EUSIPCO). Aspects of this work has also been exposed to the research community in the form of a downloadable dataset (synthetic dataset for crowd event detection, upon request) and source code of the proposed methodologies (Neuroinformatics). We provide additional information that is indicative of the overall acceptance of this work.

Moreover, this work has contributed to the successful implementation of several H2020 Secure Societies projects, including SURVANT, ASGARD, ANITA, ALADDIN). Especially for the Fast Track to Innovation (FTI) project SURVANT, a forensic investigation assistant was developed at Technology Readiness Level (TRL) 9. The work performed for this thesis has partially contributed to the implementation of the identically named tool for investigating forensic cases in heterogeneous repositories of surveillance videos.

- Publications
  - **A. Dimou**, D. Ataloglou, K. Dimitropoulos, F. Álvarez, and P. Daras, "LDS-inspired residual networks", IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 8, pp. 2363–2375, 2019.
  - **A. Dimou***, K. Karageorgos*, F. Álvarez, and P. Daras, "Implicit and Explicit Optical Flow Regularization", Sensors 2020, 20(14), 3855. (*Equal contribution)
  - D. Ataloglou, **A. Dimou**, D. Zarpalas, and P. Daras, "Fast and precise hippocampus segmentation through deep convolutional neural network ensembles and transfer learning", Neuroinformatics, vol. 17, no. 4, pp. 563–582, 2019.
  - **A. Dimou**, P. Medentzidou, F. Álvarez, and P. Daras, "Multi-target detection in CCTV footage for tracking applications using deep learning techniques", in 2016 IEEE International Conference on Image Processing (ICIP), pp. 928–932, 2016.
  - A. Psaltis, **A. Dimou**, F. Álvarez, and P. Daras, "Flow R-CNN: Flow-enhanced object detection", submitted on July 15th in ICPR 2020.
  - K. Karageorgos, **A. Dimou**, A. Axenopoulos, P. Daras, and F. Álvarez, "Semantic filtering for video stabilization", in 2017 14^{th} IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6, IEEE, 2017.
  - L. Lazaridis, **A. Dimou**, and P. Daras, "Abnormal behavior detection in crowded scenes using density heatmaps and optical flow," in 2018 26^{th} European Signal Processing Conference (EUSIPCO), pp. 2060–2064, IEEE, 2018.

- Additional info
  - Contribution to European projects:
    * SURVANT -Surveillance Video Archives Investigation Assistant (H2020-Fast Track to Innovation, GA n°720417)
    * ASGARD - Analysis System for Gathered Raw Data (H2020-Secure Societies, GA n°700381)
    * ANITA - Advanced Tools for fighting online illegal trafficking (H2020-Secure Societies, GA n°787061)
    * ALADDIN - Advanced hoListic Adverse Drone Detection, Identification and Neutralization (H2020-Secure Societies, GA n°740859)

9.3 Limitations and future work

The work on LDS-inspired ResNet has demonstrated the importance of reducing the complexity of the information modeled by the network. Using content-based modeling of the information, the reduced residual information was better modeled by the residual branch of the LDS block, leading to improved performance in visual analysis tasks. However, the LDS-inspired content-based modeling is a computationally expensive procedure limiting the applicability of the method in scenarios with low processing resources. Investigating computationally friendly processes to perform an analogous procedure is an interesting line of research for the future.

During our work on optical flow regularization, it was noticed that the accuracy of the results was highly related to the quality of the semantic information used. The available ground truth semantic segmentation is not fully aligned with the respective images, causing inconsistencies when combined. It is believed that training with finer segmentation masks would prove even more beneficial. Moreover, the use of semantic and not instance segmentation masks has limited the effectiveness of the proposed methods on the boundaries among objects of the same class. The use of instance segmentation masks is expected to further improve the results. Finally, the experimental evaluation has shown that the training scheduling is of paramount importance for optical flow estimation networks and should be further explored.

In our work on motion-enhanced object detection, the addition of a pseudo-temporal branch for the Mask RCNN architecture was proposed. The experimental evaluation showed that the

results are sensitive to the accuracy of the estimated flow. Future work includes the investigation of re-adjusting the proposed pseudo-temporal branch utilizing a more sophisticated optical flow estimation methodology. Moreover, a more systematic review of the effect of the flow branch to the main core training and the performance of the other branches is required.

Regarding multi-object tracking, a late fusion model was utilized to incorporate the information of different modalities, namely appearance, position, velocity, volume, and social interaction. The appearance model, achieves successfully to connect tracklets of the same person with a high accuracy. However, long occlusions, big changes in illumination, and people similarly dressed are still deteriorating the results. Positioning is a robust information cue but it can be deceitful in scenes with a big range in depth. While volume and velocity are not robust criteria on their own, they are complementary to the rest of the modalities and they are contributing in the fusion process. Lastly, the social interaction modality shows encouraging results especially in multi-object scenarios. However, in crowded scenarios occupancy grids may be similar even for different targets. Given the diversity of strengths and weaknesses of the utilized modalities, it is important to further study the fusion process, including also early and slow fusion methods.

Our work on video stabilization is 2D-based and it cannot provide 3D camera motion planning. The degree of stabilization, though, can be controlled by selecting the appropriate temporal support. Our method relies on the quality of optical flow calculation and image segmentation, which, as seen in figure 7.5, can identify persons unexpectedly (*e.g.*toys, posters). Temporally consistent semantic segmentation can be explored to remove such artifacts. Since we have shown that it is possible to integrate deep learning methods in the filtering stage of a stabilization pipeline, examining smoothing and result synthesis is an interesting research line. There are promising results in the field of novel view synthesis [199] and image impainting [200] that can be explored.

Crowd behavior analysis is an increasingly important topic that is not sufficiently developed due to the lack of annotated datasets. In this work, a new option to produce synthetic but realistic datasets using the GTA engine has been explored. An abnormal event detection method based on crowd density heatmaps and optical flow has been presented. The added value of training on synthetic data and its limitations in terms of performance needs to be further explored. Synthetic data can be used in different ways: augmenting real data, using transfer learning principles or even explicitly. Moreover, the extraction of crowd density heatmaps that are adjustable to specific crowd behavior applications can significantly improve their performance.

Finally, in this thesis, we have worked on improving object recognition and tracking, the building blocks of video analytics. One line of future work is related to combining those blocks to build novel video analytics applications that are focused on specific problems and needs. Many of the building blocks share a core part of feature extraction for visual content. More effort should be spent to develop a universal feature extraction framework that will be reused by multiple modules, improving efficiency. Morever, given the current trend for video processing on the edge, developing resource efficient versions of these modules is of great importance for the sector.

References

[1] MarketsandMarkets, "Video surveillance market by system, offering (hardware (camera, storage device, monitor), software (video analytics, video management system) and service (vsaas)), vertical (commercial, infrastructure, residential), and geography - global forecast to 2025." `https://bit.ly/2XgODZm`. Accessed: 2020-05-04.

[2] K. Nuechterlein, R. Parasuraman, and Q. Jiang, "Visual sustained attention: image degradation produces rapid sensitivity decrement over time," *Science*, vol. 220, no. 4594, pp. 327–329, 1983.

[3] MarketsandMarkets, "Video analytics market by type (software, services), application (intrusion management, incident detection, people/crowd counting, traffic monitoring), deployment (on-premises and cloud), vertical, and region - global forecast to 2023." `https://bit.ly/3ccNy9i`. Accessed: 2020-05-04.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[5] D. O. Gorodnichy, "Video analytics: Technology maturity, deployment challenges, and roadmap," in *Proceedings of the Third Interdepartmental Workshop on Video Technologies for National Security-VT4NS*, vol. 10, pp. 2014–36, 2010.

[6] N. Gagvani, "Challenges in video analytics," in *Embedded Computer Vision*, pp. 237–256, Springer, 2009.

[7] V. Magoulianitis, D. Ataloglou, A. Dimou, D. Zarpalas, and P. Daras, "Does deep super-resolution enhance uav detection?," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, 2019.

[8] A. Dimou, D. Ataloglou, K. Dimitropoulos, F. Alvarez, and P. Daras, "Lds-inspired residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2363–2375, 2019.

[9] D. Ataloglou, A. Dimou, D. Zarpalas, and P. Daras, "Fast and precise hippocampus segmentation through deep convolutional neural network ensembles and transfer learning," *Neuroinformatics*, vol. 17, no. 4, pp. 563–582, 2019.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[11] K. Karageorgos, A. Dimou, F. Alvarez, and P. Daras, "Implicit and explicit regularization for optical flow estimation," *Sensors*, vol. 20, no. 14, p. 3855, 2020.

[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[13] K. Karageorgos, A. Dimou, A. Axenopoulos, P. Daras, and F. Alvarez, "Semantic filtering for video stabilization," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.

[14] A. Dimou, P. Medentzidou, F. A. Garcia, and P. Daras, "Multi-target detection in cctv footage for tracking applications using deep learning techniques," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 928–932, IEEE, 2016.

[15] A. Psaltis, A. Dimou, P. Daras, and F. Alvarez, "Flow r-cnn: Flow-enhanced object detection," in *2020 ICPR - To be submitted*, 2020.

[16] L. Lazaridis, A. Dimou, and P. Daras, "Abnormal behavior detection in crowded scenes using density heatmaps and optical flow," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2060–2064, IEEE, 2018.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[21] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[22] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European Conference on Computer Vision*, pp. 646–661, Springer, 2016.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[25] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[27] T. Senst, V. Eiselein, and T. Sikora, "Robust local optical flow for feature tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1377–1387, 2012.

[28] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition," in *German Conference on Pattern Recognition*, Springer, 2018.

[29] F. Patrona, A. Chatzitofis, D. Zarpalas, and P. Daras, "Motion analysis: Action detection, recognition and evaluation based on motion capture data," *Pattern Recognition*, vol. 76, pp. 612–622, 2018.

[30] S. Liu, L. Yuan, P. Tan, and J. Sun, "Steadyflow: Spatially smooth optical flow for video stabilization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4209–4216, 2014.

[31] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[32] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3897–3906, 2019.

[33] D. S. Alexiadis, A. Chatzitofis, N. Zioulis, O. Zoidi, G. Louizis, D. Zarpalas, and P. Daras, "An integrated platform for live 3d human reconstruction and motion capturing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 798–813, 2016.

[34] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[35] D. Shulman and J.-Y. Herve, "Regularization of discontinuous flow fields," in *[1989] Proceedings. Workshop on Visual Motion*, pp. 81–86, IEEE, 1989.

[36] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 565–593, 1986.

[37] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic huber-l1 optical flow.," in *BMVC*, 2009.

[38] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.

[39] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Deep end2end voxel2voxel prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 17–24, 2016.

[40] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[41] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[42] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 6044–6053, 2019.

[43] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," *arXiv preprint arXiv:1904.05290*, 2019.

[44] T.-W. Hui, X. Tang, and C. Change Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8981–8989, 2018.

[45] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Models matter, so does training: an empirical study of cnns for optical flow estimation," *arXiv preprint arXiv:1809.05571*, 2018.

[46] M. Zhai, X. Xiang, R. Zhang, N. Lv, and A. El Saddik, "Learning optical flow using deep dilated residual networks," *IEEE Access*, vol. 7, pp. 22566–22578, 2019.

[47] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[48] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *European Conference on Computer Vision*, pp. 3–10, Springer, 2016.

[49] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Guided optical flow learning," *arXiv preprint arXiv:1702.02295*, 2017.

[50] W.-S. Lai, J.-B. Huang, and M.-H. Yang, "Semi-supervised learning for optical flow with generative adversarial networks," in *Advances in Neural Information Processing Systems*, pp. 354–364, 2017.

[51] P. Liu, M. Lyu, I. King, and J. Xu, "Selflow: Self-supervised learning of optical flow," *arXiv preprint arXiv:1904.09117*, 2019.

[52] Y. Yang and S. Soatto, "Conditional prior networks for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 271–287, 2018.

[53] J.-H. Mun, M. Jeon, and B.-G. Lee, "Unsupervised learning for depth, ego-motion, and optical flow estimation using coupled consistency conditions," *Sensors*, vol. 19, no. 11, p. 2459, 2019.

[54] I. Y. Ha, M. Wilms, and M. Heinrich, "Semantically guided large deformation estimation with deep networks," *Sensors*, vol. 20, no. 5, p. 1392, 2020.

[55] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black, "Optical flow with semantic segmentation and localized layers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3889–3898, 2016.

[56] A. Behl, O. Hosseini Jafari, S. Karthik Mustikovela, H. Abu Alhaija, C. Rother, and A. Geiger, "Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios?," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2574–2583, 2017.

[57] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *Proceedings of the IEEE international conference on computer vision*, pp. 686–695, 2017.

[58] X. Wang, D. Zhu, Y. Liu, X. Ye, J. Li, and X. Zhang, "Semflow: Semantic-driven interpolation for large displacement optical flow," *IEEE Access*, vol. 7, pp. 51589–51597, 2019.

[59] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[61] R. Girshick, "Fast r-cnn object detection with caffe," *Microsoft Research*, 2015.

[62] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[63] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[64] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, pp. 379–387, 2016.

[65] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[66] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head r-cnn: In defense of two-stage object detector," *arXiv preprint arXiv:1711.07264*, 2017.

[67] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[68] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[69] M. Najibi, M. Rastegari, and L. S. Davis, "G-cnn: An iterative grid based object detector," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[70] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[71] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[72] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[73] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[74] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, *et al.*, "T-cnn: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2017.

[75] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[76] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[77] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable, "Tracking and data association," 1990.

[78] S. Hamid Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proceedings of the IEEE international conference on computer vision*, pp. 3047–3055, 2015.

[79] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *European conference on computer vision*, pp. 28–39, Springer, 2004.

[80] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 10, pp. 1683–1698, 2008.

[81] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *European Conference on Computer Vision*, pp. 788–801, Springer, 2008.

[82] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, pp. 4705–4713, 2015.

[83] K. Loumponias, A. Dimou, N. Vretos, and P. Daras, "Adaptive tobit kalman-based tracking," in *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 70–76, IEEE, 2018.

[84] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting.," in *Bmvc*, vol. 1, p. 6, 2006.

[85] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1515–1522, IEEE, 2009.

[86] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 261–268, IEEE, 2009.

[87] P. Scovanner and M. F. Tappen, "Learning pedestrian dynamics from the real world," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 381–388, IEEE, 2009.

[88] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *European conference on computer vision*, pp. 452–465, Springer, 2010.

[89] W. Choi and S. Savarese, "Multiple target tracking in world coordinate with single, minimally calibrated camera," in *European Conference on Computer Vision*, pp. 553–567, Springer, 2010.

[90] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?," in *CVPR 2011*, pp. 1345–1352, IEEE, 2011.

[91] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European conference on computer vision*, pp. 549–565, Springer, 2016.

[92] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

[93] H. Li, Y. Li, and F. Porikli, *Computer Vision – ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, November 1-5, 2014, Revised Selected Papers, Part V*, ch. Robust Online Visual Tracking with a Single Convolutional Neural Network, pp. 194–209. Cham: Springer International Publishing, 2015.

[94] X. Zhou, L. Xie, P. Zhang, and Y. Zhang, "An ensemble of deep neural networks for object tracking," in *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 843–847, IEEE, 2014.

[95] J. Ding, Y. Huang, W. Liu, and K. Huang, "Severely blurred object tracking by learning deep image representations," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[96] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *Neural Networks, IEEE Transactions on*, vol. 21, no. 10, pp. 1610–1623, 2010.

[97] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, pp. 809–817, 2013.

[98] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *Image Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 1424–1435, 2015.

[99] Y. Chen, X. Yang, B. Zhong, S. Pan, D. Chen, and H. Zhang, "Cnntracker: Online discriminative object tracking via deep convolutional neural network," *Applied Soft Computing*, vol. 38, pp. 1088–1098, 2016.

[100] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1150–1163, 2006.

[101] B.-Y. Chen, K.-Y. Lee, W.-T. Huang, and J.-S. Lin, "Capturing intention-based full-frame video stabilization," in *Computer Graphics Forum*, vol. 27, pp. 1805–1814, Wiley Online Library, 2008.

[102] M. L. Gleicher and F. Liu, "Re-cinematography: improving the camera dynamics of casual video," in *Proceedings of the 15th ACM international conference on Multimedia*, pp. 27–36, ACM, 2007.

[103] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust l1 optimal camera paths," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 225–232, IEEE, 2011.

[104] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 78, 2013.

[105] C. Buehler, M. Bosse, and L. McMillan, "Non-metric image-based rendering for video stabilization," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2, pp. II–II, IEEE, 2001.

[106] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 44, 2009.

[107] Z. Zhou, H. Jin, and Y. Ma, "Plane-based content preserving warps for video stabilization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2299–2306, 2013.

[108] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun, "Video stabilization with a depth camera," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 89–95, IEEE, 2012.

[109] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 1, p. 4, 2011.

[110] F. Liu, Y. Niu, and H. Jin, "Joint subspace stabilization for stereoscopic video," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 73–80, 2013.

[111] A. Goldstein and R. Fattal, "Video stabilization using epipolar geometry," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 5, p. 126, 2012.

[112] Y.-S. Wang, F. Liu, P.-S. Hsu, and T.-Y. Lee, "Spatially and temporally optimized video stabilization," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 8, pp. 1354–1361, 2013.

[113] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, "Meshflow: Minimum latency online video stabilization," in *European Conference on Computer Vision*, pp. 800–815, Springer, 2016.

[114] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2720–2727, IEEE, 2013.

[115] S. Zhou, W. Shen, D. Zeng, and Z. Zhang, "Unusual event detection in crowded scenes by trajectory analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 1300–1304, IEEE, 2015.

[116] T. Xiao, C. Zhang, H. Zha, and F. Wei, "Anomaly detection via local coordinate factorization and spatio-temporal pyramid," in *Asian Conference on Computer Vision*, pp. 66–82, Springer, 2014.

[117] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 2008.

[118] V. Reddy, C. Sanderson, and B. C. Lovell, "Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, IEEE, 2011.

[119] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pp. 65–72, IEEE, 2005.

[120] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang, "Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358–368, 2016.

[121] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pp. 733–742, IEEE, 2016.

[122] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 4489–4497, IEEE, 2015.

[123] J. R. Medel and A. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks," *arXiv preprint arXiv:1612.00390*, 2016.

[124] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[125] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[126] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The microsoft 2016 conversational speech recognition system," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 5255–5259, IEEE, 2017.

[127] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[128] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, vol. 4, p. 12, 2017.

[129] A. Ravichandran, R. Chaudhry, and R. Vidal, "Categorizing dynamic textures using a bag of dynamical systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 342–353, 2013.

[130] G. Luo, S. Yang, G. Tian, C. Yuan, W. Hu, and S. J. Maybank, "Learning human actions by combining global dynamics and local appearance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2466–2482, 2014.

[131] K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Higher order linear dynamical systems for smoke detection in video surveillance applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 5, pp. 1143–1154, 2017.

[132] K. Dimitropoulos, P. Barmpoutis, A. Kitsikidis, and N. Grammalidis, "Classification of multidimensional time-evolving data using histograms of grassmannian points," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 892–905, April 2018.

[133] A. B. Chan and N. Vasconcelos, "Classifying video with kernel dynamic textures," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–6, IEEE, 2007.

[134] G. Zhou, N. Dong, and Y. Wang, "Non-linear dynamic texture analysis and synthesis using constrained gaussian process latent variable model," in *Circuits, Communications and Systems, 2009. PACCS'09. Pacific-Asia Conference on*, pp. 27–30, IEEE, 2009.

[135] K. Dimitropoulos, P. Barmpoutis, C. Zioga, A. Kamas, K. Patsiaoura, and N. Grammalidis, "Grading of invasive breast carcinoma through grassmannian vlad encoding," *PloS one*, vol. 12, no. 9, p. e0185110, 2017.

[136] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cudnn: Efficient primitives for deep learning," *arXiv preprint arXiv:1410.0759*, 2014.

[137] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, 2015.

[138] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*, pp. 630–645, Springer, 2016.

[139] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS workshop*, 2011.

[140] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning.," in *OSDI*, vol. 16, pp. 265–283, 2016.

[141] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., Computer Science, University of Toronto, 2009.

[142] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[143] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[144] X. Chen and A. Gupta, "An implementation of faster rcnn with study for region sampling," *arXiv preprint arXiv:1702.02138*, 2017.

[145] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[146] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[147] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, pp. 424–432, Springer, 2016.

[148] S. Gidaris and N. Komodakis, "Detect, replace, refine: Deep structured prediction for pixel wise labeling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5248–5257, 2017.

[149] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4733–4742, 2016.

[150] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *European conference on computer vision*, pp. 825–841, Springer, 2016.

[151] K. Li, B. Hariharan, and J. Malik, "Iterative instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3659–3667, 2016.

[152] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning: A taxonomy," *arXiv preprint arXiv:1710.10686*, 2017.

[153] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1164–1172, 2015.

[154] M. Bai, W. Luo, K. Kundu, and R. Urtasun, "Exploiting semantic information and deep matching for optical flow," in *European Conference on Computer Vision*, pp. 154–170, Springer, 2016.

[155] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3712–3722, 2018.

[156] A. D. Milner, "How do the two visual streams interact with each other?," *Experimental brain research*, vol. 235, no. 5, pp. 1297–1308, 2017.

[157] D. Ray, N. Hajare, D. Roy, and A. Banerjee, "Large-scale functional integration, rather than functional dissociation along dorsal and ventral streams, underlies visual perception and action," *Journal of Cognitive Neuroscience*, vol. 32, no. 5, pp. 847–861, 2020. PMID: 31933430.

[158] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," in *Advances in Neural Information Processing Systems*, pp. 9605–9616, 2018.

[159] K. Zafeirouli, A. Dimou, A. Axenopoulos, and P. Daras, "Efficient, lightweight, coordinate-based network for image super resolution," in *2019 International Conference on Engineering, Technology and Innovation (ICE)*, 2019.

[160] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048, 2016.

[161] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European conference on computer vision*, pp. 611–625, Springer, 2012.

[162] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[163] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3061–3070, 2015.

[164] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International journal of computer vision*, vol. 92, no. 1, pp. 1–31, 2011.

[165] A. C. Davies and S. A. Velastin, "Progress in computational intelligence to support cctv surveillance systems," *International Journal of Computing*, vol. 4, no. 3, pp. 76–84, 2014.

[166] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[167] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *CoRR*, vol. abs/1506.02025, 2015.

[168] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Framelet-based blind motion deblurring from a single image," *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 562–572, 2012.

[169] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013.

[170] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[171] H. Jin, P. Favaro, and R. Cipolla, "Visual tracking in the presence of motion blur," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 18–25 vol. 2, June 2005.

[172] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[173] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "A mobile vision system for robust multi-person tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, IEEE Press, June 2008.

[174] S. K. Sønderby, C. K. Sønderby, L. Maaløe, and O. Winther, "Recurrent spatial transformer networks," *CoRR*, vol. abs/1509.05329, 2015.

[175] H. Hill and A. Johnston, "Categorizing sex and identity from the biological motion of faces," *Current biology*, vol. 11, no. 11, pp. 880–885, 2001.

[176] R. Gao, B. Xiong, and K. Grauman, "Im2flow: Motion hallucination from static images for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5937–5947, 2018.

[177] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[178] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[179] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4340–4349, 2016.

[180] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, H. Wu, Q. Nie, H. Cheng, C. Liu, *et al.*, "Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

[181] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[182] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[183] A. Dominguez-Sanchez, M. Cazorla, and S. Orts-Escolano, "A new dataset and performance evaluation of a region-based cnn for urban object detection," *Electronics*, vol. 7, no. 11, p. 301, 2018.

[184] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[185] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3908–3916, 2015.

[186] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 300–311, 2017.

[187] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[188] X. Ma, X. Zhu, S. Gong, X. Xie, J. Hu, K.-M. Lam, and Y. Zhong, "Person re-identification by unsupervised video matching," *Pattern Recognition*, vol. 65, pp. 197–210, 2017.

[189] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Scandinavian conference on Image analysis*, pp. 91–102, Springer, 2011.

[190] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[191] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional random fields as recurrent neural networks," in *International Conference on Computer Vision (ICCV)*, 2015.

[192] Revaud, Jerome, Weinzaepfel, Philippe, Harchaoui, Zaid, and Schmid, Cordelia, "EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow," in *Computer Vision and Pattern Recognition*, 2015.

[193] I. N. Bronshtein and K. A. Semendyayev, *Handbook of mathematics*. Springer Science & Business Media, 2013.

[194] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr, "Higher order conditional random fields in deep neural networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[195] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *IEEE Intenational Conference on Computer Vision (ICCV)*, (Sydney, Australia), Dec. 2013.

[196] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *International Symposium on Neural Networks*, pp. 189–196, Springer, 2017.

[197] H. Rabiee, J. Haddadnia, H. Mousavi, M. Kalantarzadeh, M. Nabi, and V. Murino, "Novel dataset for fine-grained abnormal behavior understanding in crowd," in *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, IEEE, 2016.

[198] "Script hook v, ab software development." `http://www.dev-c.com/gtav/scripthookv/`. Accessed: 2020-09-29.

[199] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5515–5524, 2016.

[200] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," *arXiv preprint arXiv:1607.07539*, 2016.