HAL
open science

Self-supervised learning of object-centric representations with multi-object detection and segmentation

Bruno Sauvalle

To cite this version:

Bruno Sauvalle. Self-supervised learning of object-centric representations with multi-object detection and segmentation. Robotics [cs.RO]. Université Paris sciences et lettres, 2023. English. NNT : 2023UPSLM006 . tel-04106903

HAL Id: tel-04106903

https://pastel.hal.science/tel-04106903v1

Submitted on 25 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES Paris

Self-supervised learning of object-centric representations with multi-object detection and segmentation

Apprentissage auto-supervisé de représentations centrées sur les objets avec détection et segmentation multi-objets

Soutenue par

Bruno SAUVALLE

Le 8 février 2023

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Energétique

Spécialité

Informatique Temps-Réel, Robotique et Automatique

Composition du jury :

Marc Van Droogenbroeck Professeur, Université de Liège (Département d'électricité, électronique et informatique, Institut Montefiore)	*Rapporteur & Examinateur*
Mathieu Salzmann Chargé de Recherche, Ecole Polytechnique Fédérale de Lausanne (Computer Vision Laboratory)	*Rapporteur & Examinateur*
Matthieu Cord Professeur, Sorbonne Université – ISIR & Senior Scientist, Valeo.ai	*Examinateur*
Vincent Lepetit Directeur de recherche, École des Ponts ParisTech - Imagine/LIGM	*Examinateur, Président du jury*
Emilie Wirbel Robotics engineer manager, Isaac 3D perception, Nvidia	*Examinatrice*
Fabien Moutarde Professeur, Université PSL (Mines Paris, Centre de Robotique)	*Examinateur*
Arnaud de La Fortelle Professeur associé, Université PSL (Mines Paris, Centre de Robotique) & CTO Heex Technologies	*Directeur de thèse*

MINES PARIS | PSL

Abstract

Computer vision has undergone a revolution since 2012 and the application of deep learning techniques to diverse vision tasks such as image classification, object detection, image segmentation and instance segmentation. These techniques allow to get impressive results, but existing models require annotated datasets, which are costly to develop, and may be insufficient to handle rare or new events.

The goal of this thesis is to study how deep learning techniques, i.e. stochastic gradient descent and neural networks, can be used to get an interpretable representation of a scene without requiring any annotated dataset. In order to get such a representation, we consider that a scene is composed of a background and various foreground objects. We then have to be able to distinguish the background from the foreground objects present in the scene, and also to separate these foreground objects, which can touch or occlude each other.

We first study the task of fixed background reconstruction, whose goal is to build a unique background image of a scene using a short sequence of images of this scene cluttered by various objects. We address this task as a robust estimation problem, propose a new technique called background bootstrapping, which uses stochastic gradient descent, and show that it is more accurate and significantly faster than state of the art methods.

We then consider the task of dynamic background reconstruction and background/foreground segmentation. Starting from the assumption that the backgrounds of the images appearing in a video or a dataset lie on a low dimensional manifold, we are able to learn this manifold using a convolutional autoencoder. In order to improve segmentation results, we adapt the autoencoder to predict the background noise, which can be caused by turbulence, moving trees or water, and should not be considered as foreground. We then show that the proposed model is able to improve upon the state of the art for unsupervised methods on the challenging CDnet and LASIESTA benchmarks.

The segmentation of the background is a first step in order to understand the structure of a scene, but it does not allow to identify and segment the various objects appearing in a scene. In order to get a true object-centric representation of a scene, we introduce a new architecture for unsupervised object-centric representation learning, which uses attention and soft-argmax to localize each object and a transformer encoder to manage occlusions and avoid duplicate detections. We then show that this architecture is significantly more accurate than the state of the art on existing synthetic benchmarks and provide some examples of applications to real-world images taken from traffic cameras.

Résumé en français

La vision par ordinateur a subi une révolution depuis 2012 avec l'application des techniques d'apprentissage profond à diverses tâches de vision telles que la classification d'images, la détection d'objets, la segmentation d'images et la segmentation d'instances. Ces techniques permettent d'obtenir des résultats impressionnants, mais les modèles existants nécessitent des jeux de données annotés, coûteux à développer, et peuvent difficilement gérer des événements rares ou nouveaux.

L'objectif de cette thèse est d'étudier comment les techniques d'apprentissage profond, c'est-à-dire la descente de gradient stochastique et les réseaux de neurones, peuvent être utilisées pour obtenir une représentation interprétable d'une scène sans nécessiter de jeu de données annotées.

Afin d'obtenir une telle représentation, nous considérons qu'une scène est composée d'un arrière plan et de divers objets apparaissant en avant-plan. Nous devons donc non seulement être capable de distinguer l'arrière-plan de ces différents objets, mais aussi de séparer ces objets, qui peuvent se toucher ou s'occulter entre eux.

Nous étudions d'abord la tâche de reconstruction d'arrière-plan fixe, dont le but est de construire une image unique de l'arrière-plan d'une scène à l'aide d'une courte séquence d'images de cette scène encombrée par divers objets. Nous considérons cette tâche comme un problème d'estimation robuste, proposons une nouvelle technique appelée bootstrap d'arrière-plan, qui utilise la descente de gradient stochastique, et montrons qu'elle est plus précise et considérablement plus rapide que les meilleures méthodes existantes.

Nous considérons ensuite la tâche de reconstruction d'arrière-plan dynamique et de segmentation d'arrière-plan/avant-plan. À partir de l'hypothèse selon laquelle les arrière-plans des images apparaissant dans une vidéo ou un jeu de données sont situés sur une variété de petite dimension, nous sommes en mesure d'apprendre cette variété à l'aide d'un autoencodeur convolutionnel. Afin d'améliorer les résultats de segmentation, nous adaptons l'autoencodeur pour prédire le bruit d'arrière-plan, qui peut être causé par la turbulence ou les mouvements des arbres ou de l'eau. Nous montrons ensuite que le modèle proposé donne de meilleurs résultats que les meilleures méthodes non supervisées existantes sur les exigeants benchmarks CDnet et LASIESTA.

La segmentation de l'arrière-plan est une première étape pour comprendre la structure d'une scène, mais elle ne permet pas d'identifier et de segmenter les divers objets apparaissant dans une scène. Afin d'obtenir une représentation véritablement centrée sur les objets d'une scène, nous introduisons une nouvelle architecture pour l'apprentissage non supervisé de représentations centrées sur les objets, qui utilise l'attention et le soft-argmax pour localiser chaque objet et un transformer encodeur pour gérer les occlusions et éviter les doubles détections. Nous montrons ensuite que cette architecture est considérablement plus précise que l'état de l'art sur les benchmarks synthétiques existants et fournissons quelques exemples d'applications à des images réelles prises par des caméras de circulation.

Acknowledgements

I would like to warmly thank all the members of the jury for the time and attention they have dedicated to my work.

I extend my thanks to Arnaud de La Fortelle, my PhD advisor, for accompanying me during the three years of my thesis, and to Fabien Moutarde, who warmly advised me when I was looking for a laboratory likely to welcome me.

I am grateful to my wife Suzanne and my daughter Jeanne for their patience and support during this time.

I also thank Cédric Denis-Rémis, who, when he was in charge of human resources at Mines Paris, suggested me to explore the field of artificial intelligence.

My thanks also go to Sascha Hornauer, who generously provided constructive feedback on my work.

Finally, I thank Vincent Laflèche, General Manager of Mines Paris, without whom none of this would have been possible.

Contents

Contents **4**

1 Introduction **7**
1.1 Résumé en français . . . 7
1.2 Context . . . 7
1.3 Research topic . . . 8
1.4 Overview of contributions . . . 9
1.5 Experimental setup . . . 10

2 Background: representation learning **11**
2.1 Résumé en français . . . 11
2.2 Motivation . . . 11
2.3 Unstructured vectorial representations . . . 12
2.3.1 Basic vectorial representations : the concept of feature vector . . . 12
2.3.2 Contextual vectorial representations . . . 18
2.4 Basic containers for vectorial representations: feature maps, sets and graphs . . . 20
2.4.1 Feature maps . . . 20
2.4.2 Sets . . . 21
2.4.3 Graphs . . . 22
2.5 Interpretable representations . . . 23
2.5.1 Reconstruction-based structured representation learning . 23
2.5.2 Using consistency targets to build structured representations . . . 27
2.6 Motivation for studying object-centric representations . . . 27
2.6.1 Scene or video understanding . . . 28
2.6.2 Scene dynamics understanding and prediction . . . 28
2.6.3 Reinforcement learning and robotics . . . 28

3 Fixed background reconstruction **33**
3.1 Résumé en français . . . 33
3.2 Abstract . . . 33
3.3 Introduction . . . 34
3.4 Related Work . . . 35
3.5 Proposed Algorithm for Background Reconstruction . . . 37
3.5.1 Motivation . . . 37
3.5.2 Bootstrap weights . . . 38

3.5.3 Optical Flow Weights . . . 39
3.5.4 Abnormal Image Weights . . . 40
3.5.5 Management of Intermittent Motion . . . 41
3.5.6 Statement of the Optimization Problem . . . 41
3.6 Evaluation of the Proposed Model . . . 42
3.6.1 Implementation Details . . . 42
3.6.2 Evaluation on SBMnet dataset . . . 43
3.6.3 Evaluation on SBI Dataset . . . 44
3.6.4 Ablation Study . . . 45
3.6.5 Computation Time . . . 45
3.6.6 Image Samples . . . 47
3.6.7 Hyperparameter Tuning . . . 47
3.7 Conclusion of chapter 3 . . . 48

4 Dynamic background reconstruction 51
4.1 Résumé en français . . . 51
4.2 Abstract . . . 51
4.3 Introduction . . . 52
4.4 Related work . . . 53
4.5 Model description . . . 55
4.5.1 Reconstruction loss using background bootstrapping . . . 55
4.5.2 Optimized thresholding using background noise estimation 56
4.5.3 Detecting significant background changes . . . 58
4.6 Experimental results . . . 59
4.6.1 Evaluation method . . . 59
4.6.2 CDnet 2014 dataset . . . 59
4.6.3 LASIESTA dataset . . . 60
4.6.4 BMC 2012 dataset . . . 61
4.6.5 Non-video image datasets : Clevrtex, ObjectsRoom, ShapeStacks . . . 61
4.6.6 Robustness to domain shift and fine-tuning . . . 62
4.6.7 Implementation details . . . 63
4.6.8 Computation time . . . 63
4.6.9 Limitations . . . 64
4.6.10 Ablation study . . . 64
4.7 Conclusion of chapter 4 . . . 65
4.8 Appendix to chapter 4 . . . 65
4.8.1 Autoencoder architecture . . . 65
4.8.2 Additional implementation details . . . 66
4.8.3 Additional image samples . . . 67

5 Unsupervised object-centric representation learning and multi-object segmentation 74
5.1 Résumé en français . . . 74
5.2 Abstract . . . 74
5.3 Introduction . . . 74
5.4 Motivation for using attention maps and soft-argmax for object localization . . . 75
5.5 Related work . . . 77
5.6 Description of proposed model . . . 79

5.6.1 Model architecture 79
5.6.2 Model training 81
5.7 Experimental results 83
5.7.1 Evaluation on public benchmarks 83
5.7.2 Quality of learned object representations 85
5.7.3 Qualitative evaluation on real-world traffic videos 85
5.7.4 Ablation study and additional experiments 86
5.7.5 Computation time 94
5.8 Conclusion of chapter 5 94
5.9 Appendix to chapter 5 95
5.9.1 Hyperparameter values 95
5.9.2 Pseudo-code for objects encoder and decoder 95
5.9.3 Additional implementation details 95
5.9.4 Additional image samples 98

6 Discussion and conclusion 108
6.1 Résumé en français 108
6.2 Main ideas developed in this thesis 108
6.3 Limitations and future works 109
6.4 Publications 110

7 Bibliography 112

List of Figures 137

List of Tables 140

Chapter 1

Introduction

1.1 Résumé en français

Nous présentons dans cette introduction le contexte dans lequel s'inscrit cette thèse au centre de robotique de Mines Paris, l'objectif principal de la thèse, qui est d'étudier comment construire une représentation structurée et interprétable d'une scène sans avoir accès à des données annotées ainsi que les principaux résultats obtenus, qui seront développés en détail dans les chapitres suivants.

1.2 Context

This research, is conducted in the center of robotics (CAOR) of Mines Paris, under the supervision of Prof. Arnaud de La Fortelle.

The center of robotics works in the areas of autonomous vehicles, intelligent transport systems, mobile and collaborative robotics and virtual reality.

The main research domains of the center are :

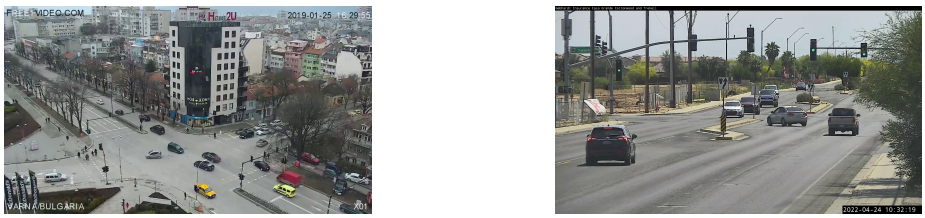
- perception and machine learning
- numerization and analysis of 3D point clouds
- virtual reality and man-machine interaction
- non linear control, advanced filtering and motion planning
- urban logistics

One can decompose the tasks performed by a robot or an autonomous vehicle in three domains, following the Sense-Plan-Act paradigm: The perception domain has to convert the outputs provided by the various sensors (camera, Lidar, etc..) into an explicit structured representation of the environment. The planning domain takes as input this structured representation of the environment and the goal or reward function provided to the system, and provides a target trajectory in order to reach the projected goal. Finally the control domain takes as input the target trajectory and computes the required commands so that the robot or autonomous vehicle follows the planned trajectory.

Deep learning methods are now widely used in the perception domain, using supervised methods, and are already implemented by car makers in commercial products. In this domain, CAOR's researchs are mainly focused on the analysis of 3D point clouds [Thomas et al., 2019, Roynard et al., 2018a, Roynard et al., 2018b]. Some deep learning models [Devineau et al., 2018] are also studied in CAOR to improve the control of vehicules, learning the inverse dynamics of a car taking into account both the longitudinal and the lateral dynamics, and showing that it provides better control commands than classical uncoupled control models in challenging driving situations.

The center for robotics is also interested in introducing deep learning methods for planning tasks. These methods could for example be used to predict the behavior and interactions of vehicles or to model using imitation learning what is a "normal" behavior on a road in a given country, considering that this behavior can be quite different from simply applying the available traffic and safety rules.

However those methods require large datasets of car trajectories. Considering the wide availability of video feeds on the Internet and produced in real time by fixed monocular traffic cameras in various cities in the world, a PhD student of the lab, A. Clausse, has built a tool allowing to extract car trajectories from these video feeds [Clausse et al., 2019]. This tool used Mask R-CNN [He et al., 2020b] for car detection and localization, and a Intersection-over-Union tracker combined with a Rauch-Tung-Striebel (RTS) smoother. However, the trajectories extracted using this tool were not robust enough for the the intended purpose and additional work remains necessary. In a similar study, Ren et al. [Ren et al., 2018], provided a quantitative estimate of the performance of a pretrained Fast-R CNN network for vehicle detection using fixed monocular traffic cameras. On a daytime dataset, the precision and recall values for a single frame were measured to be 0.57 and 0.55, respectively, which is clearly too low for applications, although smoothing using several frames allows to improve performances. The precision for the alternative SDD-VGGnet model was measured to be 0.907 and recall was measured to be 0.354.



(a) Varna (Bulgaria) **(b)** Casa Grande (USA, Arizona)

Figure 1.1: Examples of real-time traffic webcam image available on the Internet

1.3 Research topic

The goal of this thesis is to study how to build a structured and interpretable representation of a scene without access to human-annotated data.

The efficiency of vision algorithms has improved tremendously with the advent of deep learning. However existing methods suffer from three key weak-

nesses :

- Supervised learning algorithms require very large annotated datasets to be efficient. These datasets are costly to develop and to update.
- While they give very good results when the images to analyse follow the same statistical distribution as the images provided in the training set, this is not the case any more when this condition is not satisfied, for example under adversarial attack or exceptional circumstances.
- They can unpredictably make very large mistakes, which no human observer would ever make.

The approach which will be the main focus of this thesis is to consider that the data to be analyzed have a natural structure which can be exploited for unsupervised data extraction: A traffic scene image is not a simple matrix of pixels, but can often be considered as the superposition of a background image and various foreground object images, and we work under the hypothesis that this structure can be discovered in a fully unsupervised way.

We then would like to represent the background using a low dimensional latent vector z_b, and the various objects using appearance latent vectors z_k^{what}, object position coordinates and scales. The main challenges that have to be addressed in order to build this kind of representations on real-world scenes are:

- The complexity of the background and background changes in real-world scenes. Being able to accurately distinguish the foreground objects from the background is obviously an important prerequisite for any unsupervised object detection model, but is not properly addressed in existing models, which are mainly focused on object discovery.
- The low performances of existing unsupervised object detection models, which struggle to detect objects of different sizes and to manage occlusions between objects. It then seems necessary to design a completely new unsupervised object discovery architecture.

Our goal is then to build an unsupervised object detection model which is able to handle scenes with complex backgrounds, objects of any sizes, and significant occlusions between objects.

1.4 Overview of contributions

The three contributions in this thesis are the followings:

- We first consider the problem of **fixed background reconstruction**: using as input a video sequence taken from a fixed camera, our goal is to predict one image, which should the best background estimate for this video sequence. In order to address this challenge, we introduce the concept of background bootstrapping and the associated robust loss function, and show that these tools allow to improve upon the state of the art for this specific simple task.

- We then consider the more complex task of **dynamic background reconstruction and foreground/background segmentation**: using as input a video sequence taken from a camera, our goal is now to predict one background for each input image. We show that using the hypothesis that the backgrounds lie on a low dimensional manifold together with the robust loss function which has been introduced for fixed background reconstruction also allows to improve upon the state of the art on this task. To our best knowledge, this is the first time a background model is able to perform dynamic background reconstruction on videos taken from pan-tilt-zoom cameras and also on some non-video image sequences.

- We then introduce a new architecture for **unsupervised object detection and segmentation** which uses attention and soft-argmax for object localization instead of anchor grids. A transformer encoder to manage occlusions and a pretrained background model for background reconstruction. We show that this model significantly improves upon the state of the art on synthetic benchmarks and provide examples of applications to real-world traffic videos.

1.5 Experimental setup

The first research results of this thesis where obtained using a desktop PC with two 2080 TI Nvidia RTX GPU.

This research was partially funded by CARNOT contract MAIA3 1901169, which allowed to purchase in February 2021 a 4-GPU server dedicated to the project with the following specifications :

AIME A4000 server based on ASUS ESC4000A-E10 barebone
GPU: 4 $\times$ 3090 Nvidia RTX 24GB
CPU: EPYC 7402 (24 cores, Rome 2.8 Ghz)
Memory: 128 GB ECC DDR4 3200 Mhz
SSD : 2x 2.5" 4TB U.2 NVMe TLC

The codes of all the models presented in this thesis are available on the Github platform[1].

[1] https://github.com/BrunoSauvalle

Chapter 2

Background: representation learning

2.1 Résumé en français

Afin de motiver la notion d'apprentissage de représentations centrées sur les objets, nous présentons dans ce chapitre les différents types de représentations développées actuellement grâce aux techniques d'apprentissage profond. Nous examinons dans un premier temps les représentations non structurées, qui peuvent prendre la forme de représentations vectorielles (vecteurs de caractéristiques, prenant en compte ou non le contexte) ou de cartes de vecteurs de caractéristiques, et abordons la problématique de la représentations des ensembles et des graphes. Dans un deuxième temps, nous examinons les différentes techniques utilisées pour construire des représentations directement interprétables : méthodes utilisant la reconstruction comme tâche d'entraînement, permettant de construire des représentations de scènes 3D, des représentations centrées sur les objets ou des représentations hiérarchiques, et méthodes utilisant de critères de cohérence tels que la symétrie par renversement du temps ou les contraintes issues de la géométrie épipolaire afin de développer des modèles non supervisés d'estimation de mouvement, de profondeur ou de flux optique. Nous présentons finalement les différents résultats disponibles actuellement sur les applications possibles des représentations centrées sur les objets.

2.2 Motivation

In order to better motivate the concept of structured representation learning, we first provide some background on the importance of representation learning for deep learning applications.

A wide variety of deep learning algorithms have been developed in the last ten years, and it would not be feasible to review all. However representation learning has shown to be a unifying factor in both vision models and natural langage processing (NLP) models in recent years:

- For vision tasks, it has been noted since 2014 that the intermediate representations learned by networks trained for image classification could be

reused for lots of other tasks like object detection, object segmentation, panoptic segmentation, etc. with only minimal adaptation or fine-tuning. It was also noted that these pretrained network could also produce efficient representations of objects which were never part of the training set [Donahue et al., 2014, Yosinski et al., 2014, Razavian et al., 2014].

- In the NLP domain, the impressive improvements obtained in the last few years are associated to progress in representation learning, which has evolved from learning word representation vectors [Mikolov et al., 2013a] to more complex contextual representations. For example, the BERT language model [Devlin et al., 2019], whose main purpose is to learn contextual word and sentence representations, was able in 2019 to beat existing state of the art NLP algorithms on eleven different NLP tasks with a large margin, using only minimal adaptations and fine-tuning. The continuous development of large language models (LLM) since 2018 has shown that the representations generated by these models were highly efficient and could be used for a wide variety of tasks.

- Multimodal representations linking images and texts [Radford et al., 2021] have also lead to spectacular applications such as or zero-shot image classification or text to image generation [Ramesh et al., 2022].

It is then widely recognized that learning a "good" representation of complex data inputs is one of the main target of interest for deep learning research. For example, one of the leading conferences in machine learning and artificial intelligence is called International Conference on Learning Representations (ICLR).

2.3 Unstructured vectorial representations

2.3.1 Basic vectorial representations : the concept of feature vector

The most basic form of representation in deep learning is the feature vector, which can be formally described as and ordered finite list of scalar values, called features, which can be learned features or human-engineered features. The dimension of the representation is the number of scalar values and assumed to be independent of the object or data considered.

An important open question about feature vectors is to define what kind of feature vector should be considered as a "good" representation. Various lines of research have been developed :

- the simplest definition is that a good representation is a representation which can be used efficiently for downstream tasks, i.e. for prediction. For example, to evaluate if feature vectors produced by a neural network from image inputs are useful, a linear classifier is added on top of the network and trained on some image dataset. If the results are good, the representations will be considered as efficient.

- Another line of research is to consider representation learning as a form of compression or dimensionality reduction. The target should then be to get the best trade-off between compression and reconstruction loss.

- In the vision community, representation learning has also been traditionaly associated to enforcing invariance with respect to various transformations. The classical SIFT descriptors [Lowe, 2004] are optimized to provide invariance against image noise, scale changes and illumination changes. In the same way, a simple method to create representations is to optimize a deep network so that the learnt representations be invariant with respect to these transformations.

- Another approach to representation learning is that a good representation should provide a disentangled description of all the factors of variation of the dataset.

- Finaly, the development of generative models has added the requirement that the space of representations should be equipped with a probabilistic structure, allowing meaningful sampling.

We review in the following paragraphs these various lines of research:

Learning to predict and self-supervised learning. The process of getting feature vectors by training a network using labels to provide predictions can be extended to the case where no external label is provided. Various prediction tasks, sometimes called "pretext tasks" have been defined which do not need manualy annotated data, and experiments show that they lead to very efficient representations.

Let's consider for example the pretext task of predicting the orientation of an image which has been rotated by 0°, 90°, 180° or 270°. It has been shown [Gidaris et al., 2018, Kolesnikov et al., 2019, Hendrycks et al., 2019], that training a network using this pretext task in parallel with classical supervised training leads to representations with better robustness against adversarial attacks and better out of distribution detection. One can also train a network to complete an image [Pathak et al., 2016, He et al., 2022](cf Fig. 2.1), to predict the colors of an image using a black and white transform of this image [Zhang et al., 2016], or to predict the relative position of a patch in an image [Doersch et al., 2015].

Other successful methods, like deepinfomax [Devon Hjelm et al., 2019] and contrastive predictive coding (CPC) [van den Oord et al., 2018], try to predict representations of some part of an image using global representations or representations produced from other parts of an image. In order to avoid the degenerate trivial solution where the representation is a constant, these methods use constrastive losses or mutual information maximization objectives.

The pretext tasks used in the BERT model are (1) masking 15 % of the words in a sentence and asking the model to predict the masked words (2) asking the model whether two sentences are naturally consecutive sentences. A very simple pretext task which appears to be extremely efficient in the NLP domain is next word or next token prediction, leading to the development autoregressive large language models such as GPT-3.

In order to explain the success of self-supervised learning, a great emphasis has then been put on the concept of mutual information and finding mutual information estimators for self-supervised learning. It is however now understood that computing mutual information is inherently intractable for complex datasets, since it requires a number of samples exponential as a function of the mutual information, and various approaches [Ozair et al., 2019, Xu et al., 2020]

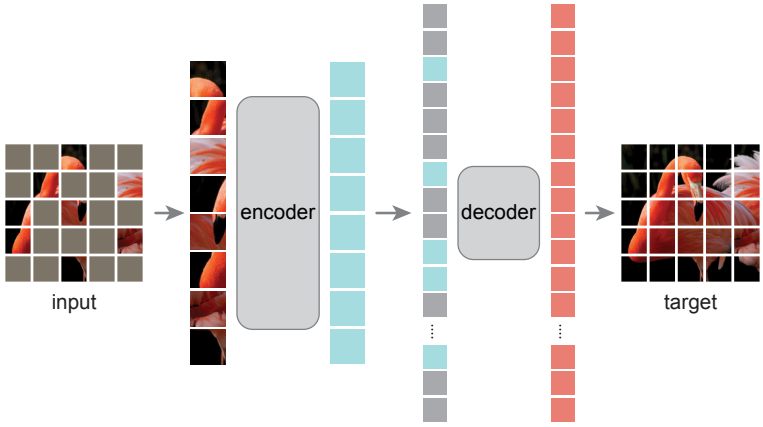


Figure 2.1: Example of state of the art unsupervised unstructured representation learning model using image completion as pretext task: MAE [He et al., 2022] is a transformer-based model which tries to reconstruct a complete image using as input an image where a large portion (e.g. 75%) of the image patches have been occluded. Source: [He et al., 2022]

have been investigated to replace the mutual information target with a more tractable target.

Unsupervised clustering seems to be an efficient pretext task to get useful representations. The difficulty with this approach is to define a meaningful loss function. In [Zhuang et al., 2019], efficient representations are obtained by asking that datapoints which belongs to the same cluster after k-means clustering of their representations should also have representations which are also close to each other.

Feature learning as a form of dimensionality reduction. It has been demonstrated [Ansuini et al., 2019] that the various layers of an image classification neural network progressively reduce the dimensionality of the data manifold. As a consequence, one could view feature learning as a form of dimensionality reduction, and autoencoders then seem to be a natural tool to get useful representations. It seems however that autoencoders using a pixel-wise L_2 loss for image reconstruction cannot lead to efficient feature learning, since they have no incentive to capture any semantic content. A successful approach using a bidirectional GAN has been implemented in [Donahue and Simonyan, 2019]: the L_2 loss is replaced with a learnt discriminator which asserts whether a pair (z, x) where z is a latent code and x an image, has been produced from a real image using an encoder (i.e. with $z = E(x)$ where E is an encoder) or has been produced from a latent code (i.e. with $x = G(z)$, where G is a generator). It is interesting to observe that this very special kind of autoencoder preserves the semantic content of the images it has to handle, although the L_2 reconstruction loss can be quite high. This method leads to performances similar to CPC without using any pretext task or any reference to mutual information targets. The representations learnt by an autoencoder can also be improved by adding to the loss function a regularization term requiring that these representations be stable with respect to various semantic-preserving transformations [Englesson

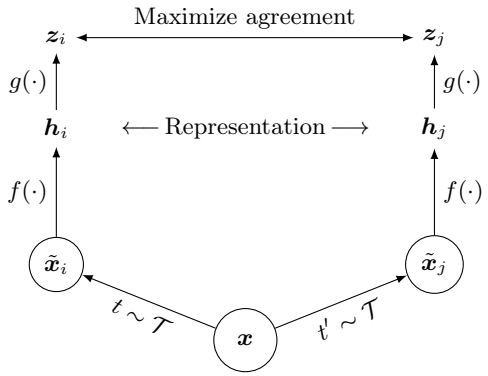


Figure 2.2: Overview of the SimCLR model [Chen et al., 2020a] "Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, the projection head $g(\cdot)$ is thrown away and the encoder $f(\cdot)$ and representation $\boldsymbol{h}$ are used for downstream tasks". Source: [Chen et al., 2020a]

and Azizpour, 2021].

Implementing symmetries and invariance. Another approach to representation learning, which has the advantage of being theorically well-motivated, is to view representation learning as a form of projection from the original dataset to the set of equivalence classes associated to symmetries or transformations which are assumed to preserve the semantic content of the data. These symmetries are usually implemented during supervised learning using data augmentation strategies or architectural choices. For example, it is well known that convolutional layers followed by max-pooling layers implement invariance with respect to translation. Getting representations which are invariant with respect to complex data transformations like small random color distortion and random Gaussian blurr is however not trivial. In order to avoid degenerate solutions, where the output of the model is a constant representation, one has to use contrastive losses, mutual information targets or specific teacher-student distillation architectures.

The SimCLR model [Chen et al., 2020a], (cf Fig. 2.2) uses a contrastive loss asking two representations to be close to each other when they are related by an elementary transformation, and to be far from each other when they were picked at random. This method allows to get better results than CPC. Ji et al. [Ji et al., 2019] use a mutual information objective to perform unsupervised clustering and self-labelling. More precisely, The model is asked to maximize the mutual information between the cluster labels associated to two input data if they are related by an elementary transformation like small Gaussian noise or random cropping.

The MoCo model [He et al., 2020a, Chen et al., 2020b, Chen et al., 2021] extends the contrastive loss computation to dictionaries larger than the mini-batch size using momentum updates. The BYOL model showed it is possible to get representations invariant to transformations without using negative samples.

Asano et al. [Asano et al., 2020] showed that combining data augmentation with self-supervised learning allowed with only one image as input to learn efficiently the first three layers of a convolutional network. Scattering networks [Bruna and Mallat, 2013, Oyallon et al., 2018], are efficient substitutes for the first layers of a convolutional neural network which do not require any learning. These two results show that the filters associated to the early layers of a convolutional neural network do not really need to be dependent from the dataset, which is consistent with the understanding that their role is to provide features which are equivariant with respect to various transformation groups. Such an approach can be generalized to pretraining full models: Baradad et al. [Baradad et al., 2021] showed that using synthetic random images with simple structures, it was possible to get pretraining performances close to the results obtained using real datasets such as Imagenet.

The idea of enforcing invariance with regards to transformation groups has been extended to more general transformations, which go beyond what one would call the natural symmetries of a problem. Misra and Van der Maaten [Misra and Van Der Maaten, 2020] showed that one could learn better representations by enforcing invariance with regard to the transformations which were used to define pretext tasks. For example, instead of asking a network to predict whether an image has been rotated or not, they ask the network to produce representations which are invariant with respect to image rotation.

Unsupervised pretraining by implementing representation invariance has been mainly successful in the vision domain, where the possible transformation groups are far larger than in the NLP domain. Some promising results [Gao et al., 2021] have however also been obtained in the NLP domain by asking the representation of a text stay invariant under the action of dropout noise.

Learning disentangled representations. It has been observed that some vectorial representations allow to perform meaningful vector calculus. For example, in the NLP domain, Mikolov noted [Mikolov et al., 2013b] that meaningful analogies could be obtained in this way with word vector representations: Adding the vector representations of "king" and "woman" and substracting the representation of "man" gives a vector which is very close to the word representation of "queen". For image representations, it has also been noted [Radford et al., 2016] that the latent space generated by deep convolutional generative adversarial networks allows to perform some latent space vector artithmetic. For example, adding the representation of a smiling woman and a neutral man and substracting the representation of a neutral woman naturally leads to the representation of a smiling man. It is also possible to get meaningful morphing of one image to another by latent space interpolation. The same kind of properties where obtained in Generative Query Networks [Ali Eslami et al., 2018], where representations of a complex 3D scene are built using partial views of this scene.

The concept of disentangled representation adds another requirement which is that a "good" representation should not only allow to perform meaningful vector calculus and latent space interpolation, but also identify and separate the various generative factors of a dataset, so that changing one latent variable should lead to a simple and meaningful transformation of the data.

It has been observed that variational autoencoders [Kingma and Welling,

2014] naturally produce disentangled representation. Modifying variational autoencoders by increasing the penalty associated to the KL divergence term seems to lead to better performance in disentangled representation learning [Higgins et al., 2017]. Another way to get generative factors is the InfoGAN model [Chen et al., 2016], which is an adaptation of the GAN [Goodfellow et al., 2020] model where the GAN image is produced from random noise vectors and random latent codes and the model is optimized so that the output image has a high mutual information with the latent codes. However a study [Locatello et al., 2019] showed that disentangled representation learning is an ill-posed problem and that the good results shown by these different models where questionable, because the authors of the models usually fine-tune the hyperparameters of their models using their prior knowledge of those generative factors. As a consequence, these models cannot really be used to discover generative factors in a completely unknown dataset.

A more principled approach to disentangled representation learning is provided by non linear independent component analysis (ICA), where the problem of identifiability has been studied for more than ten years. Khemakhem et al. [Khemakhem et al., 2020] have shown that identifying the independent generative factors of a dataset was theorically possible if the generative factors follow a factorized prior that is conditioned over an additional observed variable such as a class label or any other observation. An application [Sorrenson et al., 2020] of this approach is the discovery of 22 broadly interpretable latent variables as generative factors in the EMNIST dataset, using the labels associated to each digit. Horan et al. [Horan et al., 2021] have recently shown that this identification is also possible under the assumption that the true generative latents have a non-Gaussian distribution and that the mapping from the latents to the data is a local isometry.

Representation learning and generative models. The development of generative models has added new understanding to what should be considered a good representation: A representation model should not only be approximately injective, i.e. allow to reconstruct the data using its representation, but also surjective: Any representation vector lying in the representation space should correspond to some likely data sample. This condition is not satisfied by deterministic image autoencoders, since they allow to encode any image as a low dimensional vector, but offer no guarantee that using the generator of the autoencoder will transform any low dimensional code into a realistic looking image. This issue led to the development of variational autoencoders, which allow to generate samples using any low dimensional code sampled from a multivariate Gaussian distribution. More recently, Saseendran et al. have shown [Saseendran et al., 2021] that adding a regularizer to a deterministic autoencoder enforcing that the distribution of the latent codes follows a Gaussian (or mixture of Gaussians) distribution using an adaptation of the Kolmogorov-Smirnov test also allows to perform efficient sampling. GANs [Sohl-Dickstein et al., 2015], suffer however from the fact that, although they provide a way to generate data samples from random inputs, they are not usually able to directly recover latent codes from an input image, and cannot be considered as representation models, although it has been shown that the latent space of GAN models such as styleGAN is highly disentangled and useful for image editing. A

similar issue arises with diffusion models [Ho et al., 2020], although a specific class [Song et al., 2021] of diffusion model is deterministic and invertible and diffusion autoencoders [Preechakul et al., 2022] allow to define a meaningful latent space.

2.3.2 Contextual vectorial representations

One of the major advances in representation learning over the last few years has been the introduction of contextual representations. Intuitively, the contextual representation of an object is a representation of this object which takes also into account the context where the object lives. The concept has proved its value in the natural language processing domain, where replacing individual word representations like GloVe and WordtoVec with contextual word representations produced by BERT or XLnet [Yang et al., 2019] has lead to very significant efficiency gains. It is indeed clear that the same word can have very different meanings, depending on the context where it appears. It should however be noted that the concept of contextual representation is not really clear from a formal point of view. A single contextual representation is not a local representation and should be considered as a global representation of a sentence or a scene since it takes into account all the elements lying in the sentence or in the scene. A sequence of contextual representations could then be considered as a form of global representation. The only formal specificity of this kind of representation compared to an unstructured global representation seems that it is translation equivariant, i.e. that a shift in the input sequence or image leads to the same shift in the associated contextual representations.

The main tools used for contextual representation learning in the NLP domain are now the self-attention mechanism and the transformer model [Vaswani et al., 2017] (Fig. 2.3). Self-attention has shown spectacular results in the NLP domain [Alec et al., 2019, Brown et al., 2020], but its application in the vision domain [Ramachandran et al., 2019] is less straighforward, considering that its computational load is proportional to the square of the number of input tokens, which is clearly a problem for high definition images, which cannot be directly processed pixelwise by a transformer and have to be cut in patches which are later projected and processed by a transformer encoder [Dosovitskiy et al., 2021]. Convolutional networks can be understood as already performing a form of context aggregation between neighboring pixels, but tackling long-range dependencies with CNN remains a challenge, leading to variants like dilated convolution layers [Yu and Koltun, 2016].

However it has quickly become a standard practice in vision applications before the development of full transformer-based models to "enrich" the representation of an object using representations of related objects using an attention mechanism. For example, Yuan et al. [Yuan et al., 2020] improve the efficiency of their segmentation model by supplementing a pixel representation with a new representation computed using an attention mechanism taking into account the similarity of the considered pixel representation with the average representations of each object region. In Wang et al. [Wang et al., 2019b], the representation of one point belonging to a 3D point cloud is fused with the representation of its k nearest neighbors to get a contextual representation to perform semantic segmentation. A similar approach has been proposed [Lu et al., 2020] to enrich the representation of a frame in a video sequence: correlation weights between this

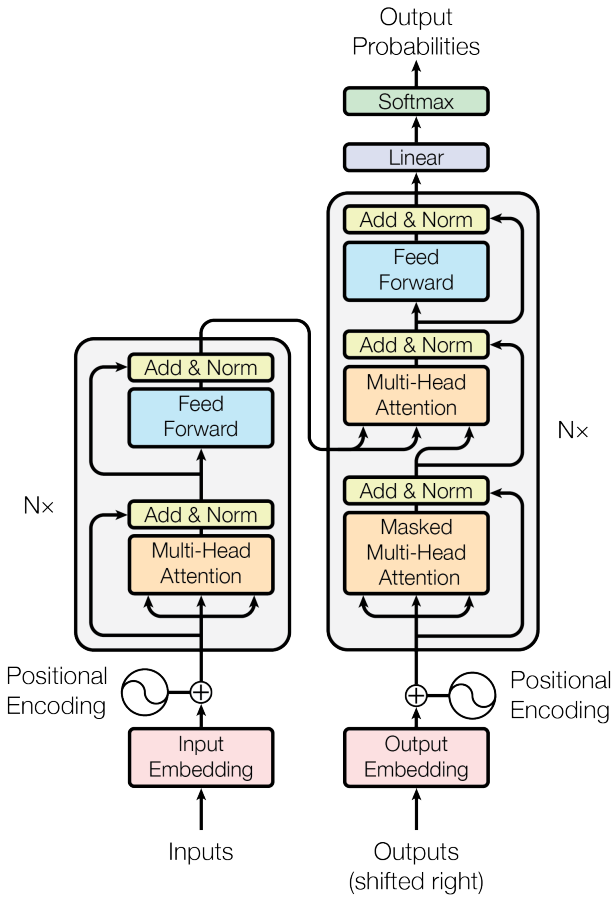


Figure 2.3: Transformer architecture: A transformer is composed of an encoder (left) containing self-attention and fully connected layers, and a decoder (right) containing masked self-attention, cross-attention and fully connected layers. A transformer encoder will be used in the model described in Chapter 5. Source: [Vaswani et al., 2017]

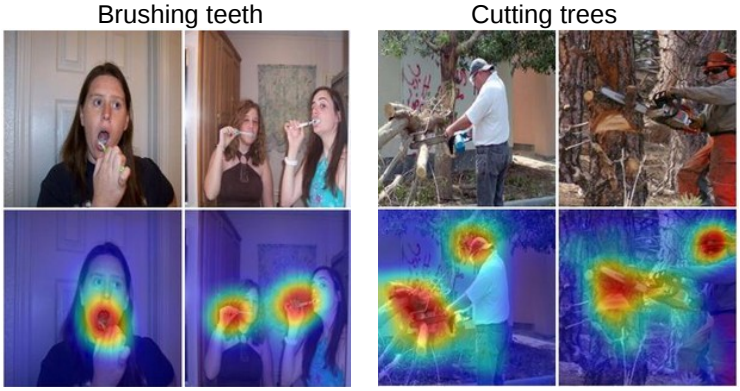


Figure 2.4: Example of class activation maps extracted from a convolutional network. Source: [Zhou et al., 2016]

frame and frames located in other parts of the video are computed and used to build a contextual representation which is the concatenation of the frame representation and the weighted sum of the representation of the other frames using the correlation weights.

2.4 Basic containers for vectorial representations: feature maps, sets and graphs

2.4.1 Feature maps

In vision applications, the various intermediate representations produced by a convolutional network are usually called feature maps, because each point of this map can be considered as the contextual feature vector associated to a particular location in the image. This interpretation leads to very interesting applications for example in object detection: in order to detect or track an object in some area of an image, one will simply look at the feature vectors lying in this area. In order to build a representation of an object covering some area of an image, one will take the average of all the feature vectors lying in this area. Class activation maps [Zhou et al., 2016] can be built using the last feature maps of a convolutional network trained for image classification and allow to localize class-specific regions which associated to an image (Fig. 2.4).

Feature maps can give rise to feature pyramids [Li et al., 2019] when several feature maps of varying scale granularity are considered, which allows to detect both small and large objects using the same detection heads.

Efficient feature maps can be produced by convolutional networks, but also by vision transformers with small patch sizes [Dosovitskiy et al., 2021, Touvron et al., 2020, Touvron et al., 2021, Touvron et al., 2022, Caron et al., 2021] (Fig. 2.5). The concept of feature map can naturally be extended to 3D maps for video data, or to 1D maps for sequential data.

A productive line of research developed in the last few years was to study how the distribution of the local values of a feature map associated to one image can be interpreted and used. It has been recognized since 2016 [Gatys et al.,

Figure 2.5: Example of multi-heads attention maps extracted from the last layer of a self-supervised VIT-S model [Caron et al., 2021]

2016] that the second moments of a feature map, i.e. the matrix of correlations between different local channel activations associated to one image, could be interpreted as a representation of the "style" of this image, so that optimizing another image to follow the same correlations between channels allows to perform style tranfer from one image to another image. Straighforward renormalization of the mean and variance of each channel map of an image even allows to get real time style transfer [Huang and Belongie, 2017].

This approach has been expanded and developped and is now so efficient that style-based methods are the current state of the art in image generation: StyleGAN [Karras et al., 2019, Karras et al., 2020, Sauer et al., 2022] first generates the style statistics associated to various layers of a convolutional networks, then generates an image following these style statistics using instance normalization.

In the same spirit, the squeeze and excitation layer [Hu et al., 2020] extracts the first order moments of a feature map, one for each channel, and use these data to modulate the channel values of the feature map, allowing some global context information to be inserted between CNN local aggregation layers.

2.4.2 Sets

The requirement to handle unordered sets is quite obvious in applications related to object detection. Objects appearing in a scene usually do not show any natural ordering, and in the same way that convolutional layers are translation-equivariant operators acting on feature maps, it is natural to require that transformations between representations of sets be equivariant with respect to the full permutation group.

Qi et al. [Qi et al., 2017] showed that a function invariant with respect to the permutation group could be approximated by the composition of a per-item transformation, applied to each elements of the set, then a fully symmetric aggregation function such as sum or max-pooling, and finally any function, and applied this model to the analysis of point clouds. The theorical analysis of maps which are equivariant with respect to the permutation group has been

performed by Zaheer et al. [Zaheer et al., 2017], which leads to a description of possible multilayer neural architectures as stacks of equivariant maps. As an example, linear maps equivariant with respect to permutations can be described as linear combinations of the identity matrix and the matrix whose coefficients are all equal to 1.

This approach was further generalized to sets composed of features maps (to anayse for example sets of images). In this case, one has to handle both the translation symmetry of the feature maps and the permutation symmetry of the set structure. The associated equivariant maps have been studied and described by Maron et al. [Maron et al., 2020].

Transformer encoders are also naturally permutation equivariant models if no positional encoding is introduced and can be used to efficiently transform set representations if the size of the set is not too high, considering that they have a quadratic complexity as a function of the number of elements in the set.

Despite these advances, machine learning with sets remain a challenge, considering that deep learning libraries naturally manage vectors and tensors, and not unordered sets. As an example, in order to check whether two sets of vectors $\{x_1, .., x_n\}$ and $\{y_1, ..., y_n\}$ are close or not, one usually computes a bipartite matching beween the x_i and y_j which requires a computation time of $O(n^3)$ [Edmonds and Karp, 1972] with the Hungarian algorithm. Set prediction is also tricky when the output has to be written as a tensor if no natural ordering or indexing of the elements of the set is available. Some authors [Locatello et al., 2020] consider that this issue can be handled by introducing some randomness, i.e. that the ordering of the predicted elements should be randomly distributed. The most common approach in object detection is however to define this ordering according to the location of the objects, either using a two dimensional grid [Girshick, 2015, Redmon et al., 2016] or some learnt positional encoding codes [Carion et al., 2020]. Another possible solution is to assign to a RNN the task of discovering such an ordering [Vinyals et al., 2016, Ali Eslami et al., 2016]. An original approach [Zhang et al., 2019] to the problem of set prediction is to consider this task as the inverse of set aggregation and approximate this inverse iteratively using the gradient of a permutation invariant function.

2.4.3 Graphs

Representing a scene as a graph showing all the objects of the scene and describing all the relations between those objects is a natural goal in computer vision (Fig. 2.6).

The development of neural network layers able to handle graph data efficiently has first been inspired by the idea of defining graph convolutions [Kipf and Welling, 2017], but is now mainly associated to the study of the maps which are compatible with the associated natural symmetries of a graph, i.e. we require that the neural transformation layers should be equivariant with respect to a relabeling of the edges or nodes, and that the aggregation layers should be invariant with respect to those relabelings [Herzig et al., 2018]. For example Xu et al. [Xu et al., 2019a] show that the most expressive form of a graph neural network can be described as a sequence of node representation updates, where the hidden representation of a node at each step is updated with the current representation of the node and the sum of the current hidden representations of the neighbouring nodes, and proposes to use a two-layers MLP

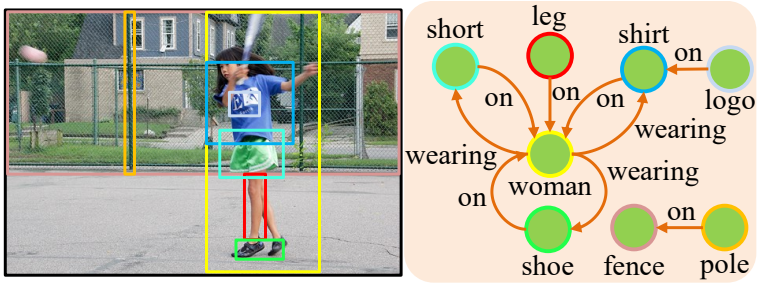


Figure 2.6: Example of scene graph. Source: [Teng and Wang, 2022]

to perform this update. The iterative updates of a graph neural network are formally very similar to the message passing updates used in graphical models and are then naturally used to perform the same kind of tasks. In order to modulate the weights of the node neighbors without breaking the equivariance property, various approachs based on attention or self-attention have been proposed [Veličković et al., 2018, Nguyen et al., 2022].

Applications of graphs to vision tasks are not limited to scene graph generation. Inference on graphs can also be used to provide a model of the interactions between moving objects and help to forecast trajectories taking into account those interactions [Kosaraju et al., 2019], or to reasoning about relations between these objects for visual question answering (VQA) applications [Hudson and Manning, 2019].

It should be also noted that the structure of graphs can be very diverse : For some graphs, both nodes and edges are equipped with vectors representations, which induces a natural bipartite structure to the graph representations updates [Xu et al., 2017]. Other graphs can have naturally weighted edges to model the strenght of the relationships. Graph nodes are not necessarily associated to the presence of an object. For example, in differentiable scene graphs [Raboh et al., 2020], nodes are associated to the possible presence of objects for each region of interest, whitout having to perform any non-differentiable decision on the presence or class of an object in this regions.

2.5 Interpretable representations

2.5.1 Reconstruction-based structured representation learning

We have seen that it is possible using self-training methods to get useful representations for downstream tasks, i.e. representations which are more compact, requires less labelled data for supervised training, or are more robust to out-of-distribution events or adversarial attacks. These representations are however not directly interpretable without any labeled data. For example, a classical deterministic or variational autoencoder trained on a large dataset will be able to represent a large image as a low dimensional vector, but the only way to guess what the values of those vectors mean is to perform some experiments or additional supervised training.

It is however possible to get interpretable representations without using la-

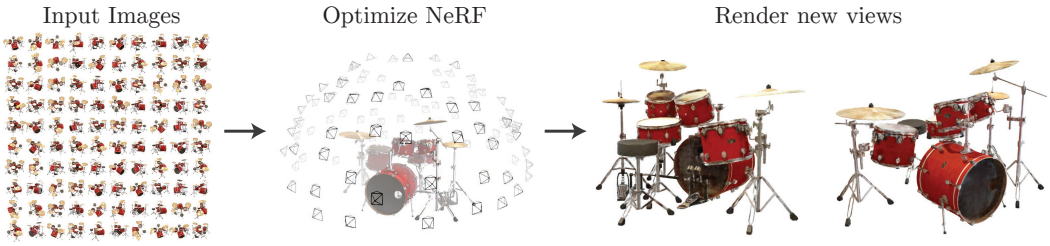


Figure 2.7: Example of application of a NeRF model. Source: [Mildenhall et al., 2020]

beled data if we put some interpretable constraints on the reconstruction process.

3D scene representation learning

As an example Chen et al. [Chen et al., 2019c] propose to use a fully differentiable renderer as a decoder. This renderer can be inverted with deep learning techniques thanks to the process of amortized inference: instead of iteratively trying to find the latent codes associated to one given image by gradient descent using this image alone, a process which is known to be highly unstable and difficult, amortized inference tries to optimize a neural network mapping any image to its generating codes using batches of synthetic images and stochastic gradient descent. Using this approach the authors are able to train a network which takes as input the synthetic image of an object and produces as output the 3D mesh associated with this object and the associated colors of the nodes. The latent representation of a scene does not need, however to be represented by a sequence of mesh points and associated color or texture values, which are the classical inputs of graphical renderers. For example, Sitzmann et al. [Sitzmann et al., 2019] propose to represent a scene by a function $\Phi : \mathbb{R}^3 \to \mathbb{R}^n$ which maps any point in the ambient space of the scene to a feature vector, for example the color and surface reflectance at this point. This function is represented by a neural network.

Multiple views of a scene can also be used to build an explicit 3D model of the scene. Neural radiance fields networks (NeRF) [Mildenhall et al., 2020] (Fig. 2.7), are neural networks which predict for all points and ray direction of a scene the associated color and density. An elementary differentiable ray-tracing rendering engine encapsulates this network, and the whole model is trained on a few number of photos of a scene. After training, the model allows to produce an accurate rendering of the same scene from any point of view.

Object-centric representation learning

In order to get an interpretable structured representation, the decoder does not need to be fully explicit. It can be a neural network with a specific architecture implementing some a priori knowledge. For example, in Detlefsen et al. [Detlefsen and Hauberg, 2019], the decoding phase is decomposed in two steps: Some latent variables coding for the shape of the considered object are first used to generate this shape. Then other latent variables defining the location, orienta-

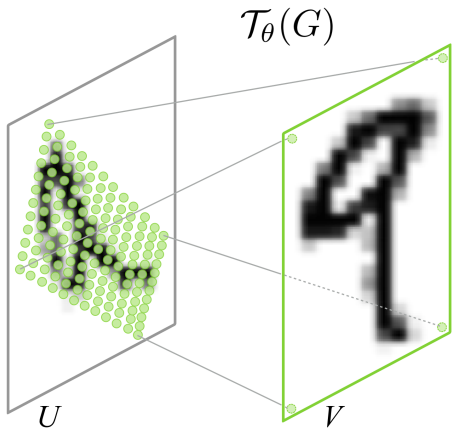


Figure 2.8: Example of image warping using a spatial transformer network. Source: [Jaderberg et al., 2015]

tion and scale of the object are used to put the shape into the correct position using a spatial transformer network [Jaderberg et al., 2015](Fig. 2.8). The architecture of the encoder is also consistent with this latent variable structure: The latent variables defining the location, orientation and scale are first computed, then used by another spatial transformer network to put the objet back at a reference position, orientation and scale so that its general shape can be coded by the encoder without being dependent on its position. In this way, it is possible to get an explicit decomposition of the latent codes in shapes-related codes and pose-related codes, and the pose-related codes are fully interpretable because the spatial transformer network is explicitly defined and does not depend on any unknown parameter.

Other latent codes decomposition can be investigated. For example, an interesting way to segment a salient object in a scene is to consider that the pixels associated to the background can be generated independently, from the pixels associated to the foreground objects [Chen et al., 2019b,Katircioglu et al., 2021]. More generally, a natural way to implement the fact that various objects in a scene are independent from each other is to build a generative model where each object has its own latent codes independent from the latent codes of the other objects.

Combining those two approaches allows to perform object detection and localization in a purely unsupervised way and leads to unsupervised learning of object-centric representations. In Eslami et al. [Ali Eslami et al., 2016], the authors propose to perform unsupervised object detection on synthetic images without background by inserting a RNN in the encoder and asking it to generate a sequence of codes of the form $(z_{where}, z_{what}, z_{pres})$, one for each object detected. z_{what} codes for the appearance of the object, z_{where} for the pose of the object, and z_{pres} indicates whether the object is present or not, so that the RNN sequence generation stops as soon as a $z_{pres} = 0$ is produced. The reconstruction is done by generating the shape of the object using z_{what}, then putting it at the right pose using a spatial transformer network with the z_{where} code, and finally by adding the generated images of all the objects together. This approach has been further developed in the models SPAIR [Crawford and

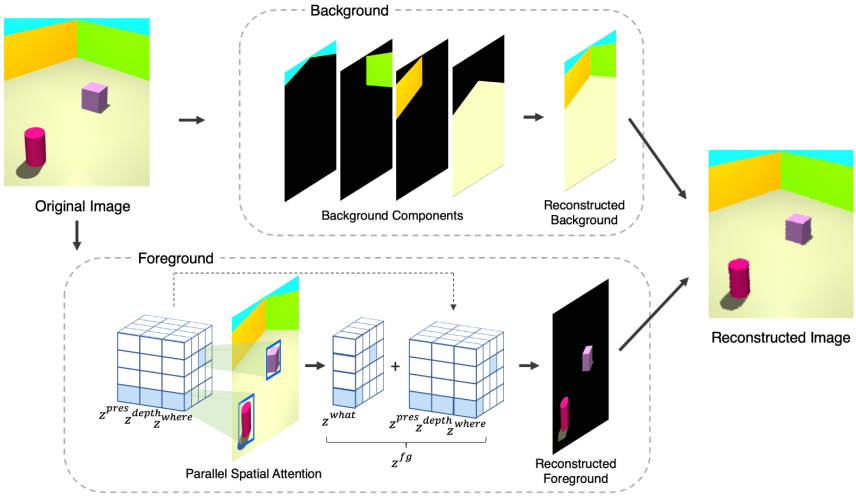


Figure 2.9: Overview of the SPACE model. Source: [Lin et al., 2020]

Pineau, 2019] and SPACE [Lin et al., 2020] (Fig. 2.9) by replacing the RNN with a grid-based object detector inspired by supervised object detectors (Fast R-CNN or YOLO).

Reconstructing an image using the sum of the images of the objects appearing in this image is clearly not satisfying considering that the occlusions cannot be handled in this way. Crawford et Pineau [Crawford and Pineau, 2019] and Lin et al. [Lin et al., 2020] manage this problem by requiring the model to predict also the mask of the object using z_{what} and to encode the depth of the objects with a new latent variable z_{depth}. The unsupervised object detection model which will be presented in chapter 5 implements a similar approach.

Hierarchical representations

Another approach which can be used to obtain more interpretable representations is to assume that the distribution of a dataset can be represented by a probabilistic model which is organized as a Bayesian network. To be consistent with this probabilistic model, the associated encoders and generators have to follow a sequential or hierarchical generation or encoding process. Although autoregressive models used in the NLP domain do not lead to interpretable representations, hierarchical models used for image applications often produce structured representations which can be considered as interpretable. For example it has been observed in the StyleGAN model, which uses an intermediate latent space coding for the styles of the image, that the first style latent variables are responsible for the high-level abstract content of the data such as shape and pose whereas the last style latents code for appearance details such as hair color (Fig. 2.10).

The latent variables learnt in hierarchical variational autoencoders [Child, 2020] have also been shown to have a structure: The top level latents code for low frequency features, while the low level latents code for small details (high frequency) appearing in the image. A possible application of this kind of

representation is image super-resolution [Prost et al., 2022] .

2.5.2 Using consistency targets to build structured representations

Extracting interpretable representations from data can also be done without using a reconstruction target: Implementing consistency targets using some a priori or expert knowledge about the data or the tasks which have to be handled in a learning-based model can also lead to useful structured representations.

This a priori knowledge can appear to be trivial, but is sometimes sufficient to allow for efficient self-supervised training. Let's for example consider the task of tracking a moving object on a video using the feature vector associated to this object and a Siamese network [Bertinetto et al., 2016]. It is natural to require that if such a model tracks an object A from some point x at time t to some other point x' at time t', then the same model applied to the reverse video should track the object A from the point x' at time t' back to the point x at time t [Wang et al., 2019a]. This kind of requirement can be implemented by defining a consistency loss and performing the usual stochastic gradient descent on this consistency loss, and is surprisingly sufficient to get a reasonable tracking model and useful feature vectors for the detected objects.

For vision tasks on 3D scene using multiple views, the required a priori knowledge can be provided by epipolar geometry. For example, Bian et al. [Bian et al., 2019] build a model which can predict ego-motion and a depth map using sequential pairs of monocular consecutive frames. The consistency loss is built using the fact that using predicted ego-motion and depth map of the first frame, it is possible to predict what the next frame will look like with geometric computations, and optimizing the associated consistency loss is enough to self-train the model without any human supervision. Epipolar geometry constraints are also used in [Zhong et al., 2019] to improve unsupervised optical flow computations when the objects appearing in the scene are all rigid bodies.

2.6 Motivation for studying object-centric representations

The goal of this thesis is to study how deep learning techniques, i.e. stochastic gradient descent and neural networks, can be used to get an interpretable representation of a scene without requiring any annotated dataset. We will build in this thesis two kind of representations of a scene:

- unsupervised object segmentations
- unsupervised object-centric representations.

Unsupervised object segmentations are fully interpretable, and can be used without further treatment for applications such as tracking or video surveillance. Object-centric representations are structured representations, but further treatments are necessary to use them. We observe that object-centric representations align well with the way humans analyse a scene and reason about it. As a consequence, it can be considered as a first step towards building neurosymbolic representations, bridging the gap between deep learning methods and symbolic

methods, and it seems reasonable to expect that object-centric representations can be useful for all tasks involving some form of reasoning about the content of a scene.

This point cannot however be considered today as a proven fact, and remains the subject of scientific debate. Due to the limitation of existing unsupervised object-centric representation models, experiments on this subject have up to now been limited to very simple scenes showing geometric shapes and uniform textures. We provide below a short overview of these experiments.

2.6.1 Scene or video understanding

The use of object-centric representations has lead to significant improvements on visual question answering (VQA) tasks involving multi-object images or videos. The ALOE model [Ding et al., 2021] (Cf Fig. 2.11) handles video and first uses a MONet model [Burgess et al., 2019] to generate an object-centric representation from each frame. These object-centric representations are first fine-tuned using self-supervised learning on the video sequence using masked token prediction, and the model is fine-tuned again on the VQA downstream task using supervised learning. This model allows to get substantial gains on various VQA datasets compared to models which do not use an object-centric approach, including models using a symbolic engine. Examples of ALOE model predictions on a video are provided in Fig. 2.12.

2.6.2 Scene dynamics understanding and prediction

Switching from pixel space representations to object-centric representations lead to a significant dimensionality reduction, and object centric representations also allow to introduce a powerful inductive bias which is that all objects should be handled in the same way, so that models managing object representations, object interactions or object manipulations should be equivariant or invariant with respect to any permutation of the object representations. As a consequence, several papers have shown that object-centric representations are particularly well suited to learn structured world models and perform future frame predictions on videos showing interacting objects [Hsieh et al., 2018, Creswell et al., 2021, Goyal et al., 2021, Min et al., 2021, Assouel et al., 2022]. For example, Fig. 2.13 provides a sample of future frame prediction from the ODDN model [Tang et al., 2022].

2.6.3 Reinforcement learning and robotics

It is widely believed that using object-centric models should allow to learn better policies on downstream tasks involving object manipulation or localization. Several object-centric reinforcement learning models have been proposed on simple multi-object scenes and have been shown to obtain better results than models handling pixel-space representations when a significant number of objects are present in the scene [Janner et al., 2019, Veerapaneni et al., 2020, Zadaianchuk et al., 2021]. Heravi et al. [Heravi et al., 2022] compare the efficiency of MoCo representations [He et al., 2020a] and object-centric representations obtained from the Slot attention model [Locatello et al., 2020] for robotic manipulations tasks involving several objects and conclude that object-centric representations

offer significant improvement and are more sample efficient compared to MoCo representations.

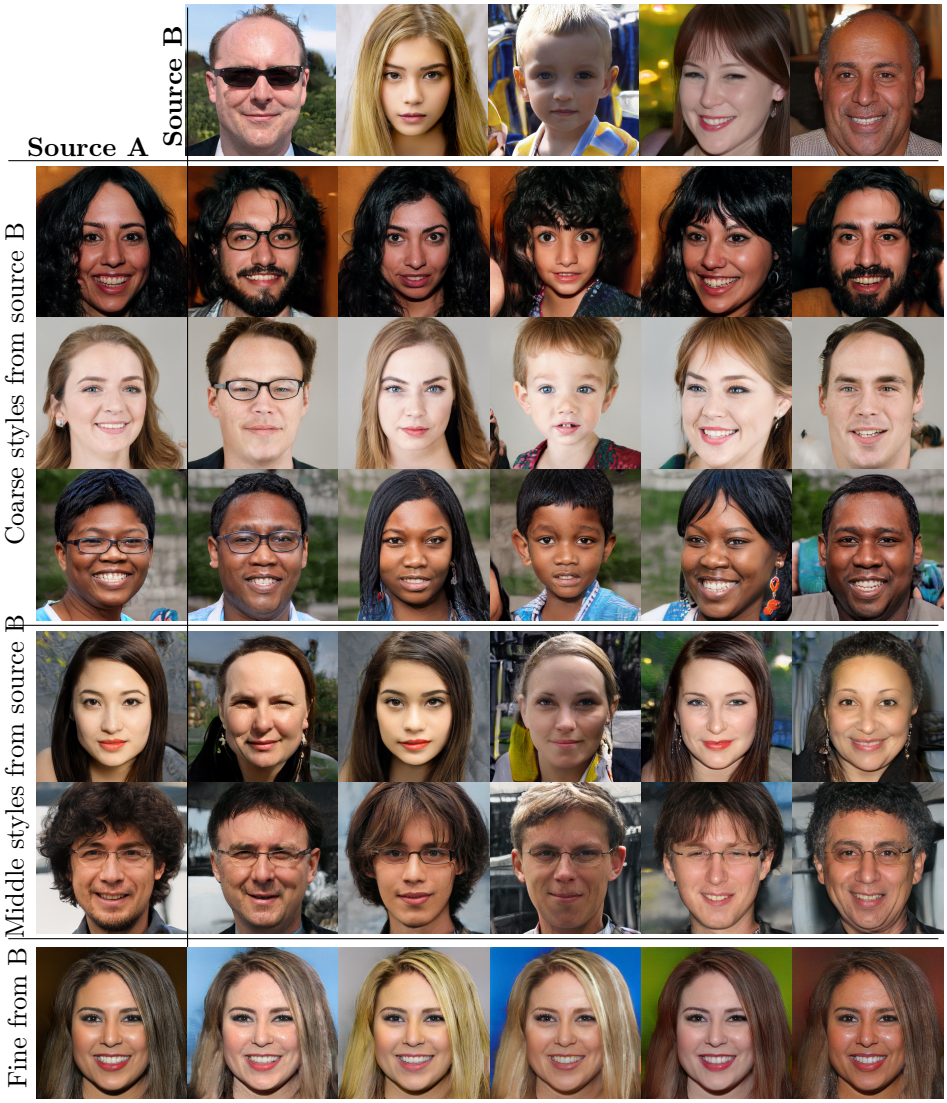


Figure 2.10: Two sets of images (Source A: first column and Source B: first row) are generated using StyleGAN. Their style latent codes are then mixed and used to generate the other images which are shown in this figure. When the top level latent codes from source B are used (row 2-4), high-level content of the source B images is transferred. However when middle (row 5-6) or low level (row 7) codes are used, only low level appearance features of source B images are transferred. Source: [Karras et al., 2019]

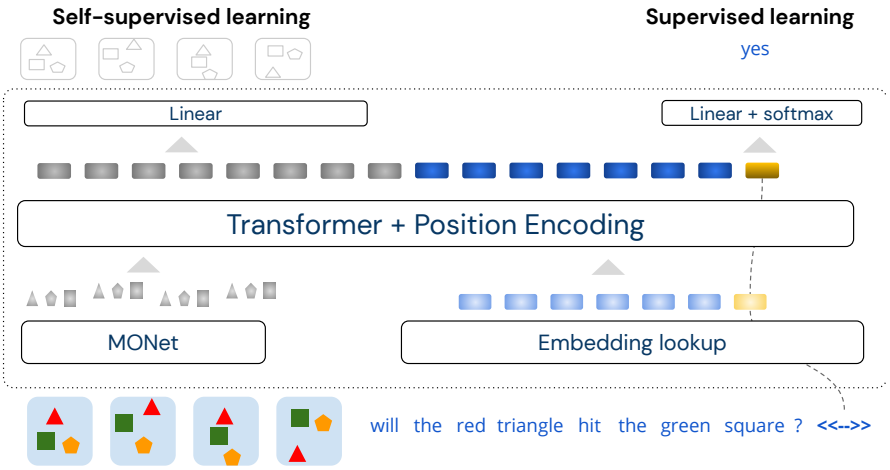
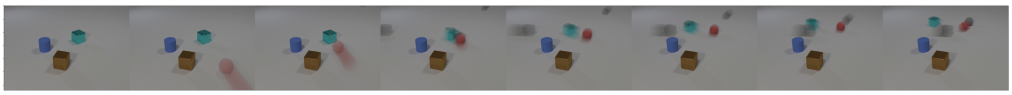


Figure 2.11: Overview of the ALOE model [Ding et al., 2021] which performs visual question answering using object representations. A transformer encoder takes as inputs (1) the object feature vectors produced by an unsupervised object detection model (MONet), (2) the embeddings of question words, (3) a CLS token. The transformed value of the CLS token is passed through an MLP to generate the final answer. Source: [Ding et al., 2021]



Q: Are there any moving brown objects when the red object enters the scene?
Model: no
Label: no

Q: What will happen next?

1. The gray cube and the brown object collide
2. The red object collides with the rubber cube

Model: 2
Label: 2

Q: If the cylinder is removed, which of the following will not happen?

1. The gray cube and the brown cube collide
2. The red object and the cyan object collide
3. The red sphere and the rubber cube collide
4. The cyan object and the brown cube collide

Model: 1, 4
Label: 1, 4

Q: How many rubber objects are moving?
Model: 3
Label: 3

Figure 2.12: Example of input video and associated ALOE VQA predictions on CLEVRER dataset. Source: [Ding et al., 2021]

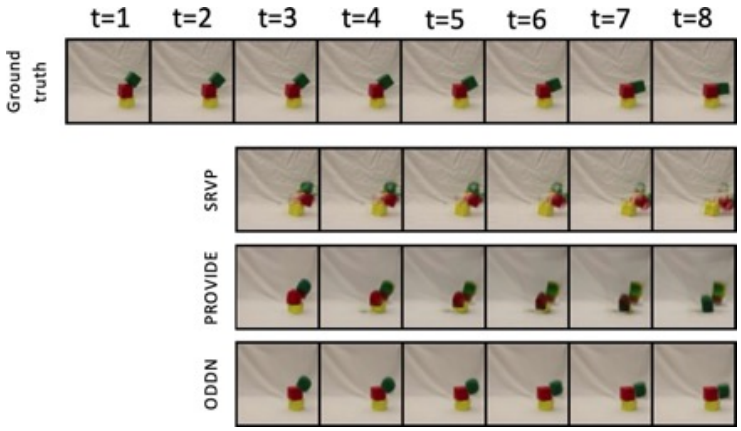


Figure 2.13: Example of future frame prediction of the ODDN model. Source: [Tang et al., 2022]

Chapter 3

Fixed background reconstruction

3.1 Résumé en français

Le but de la reconstruction d'arrière-plan, aussi appelée génération d'arrière-plan, et de reconstruire l'arrière-plan d'une scène à partir d'une séquence d'images de cette scène encombrées par divers objets en mouvement. Cette tâche est fondamentale en analyse d'image et constitue généralement une première étape préalable à des traitements plus avancés. Elle est cependant difficile car il n'y a pas de définition formelle de ce qui doit être considéré comme appartenant à l'arrière-plan ou à l'avant-plan, et les résultats peuvent être sévèrement affectés par une variété de difficultés telles que les changements de luminosité de l'image, les mouvement intermittents des objets, ou un encombrement élevé de la scène par de nombreux objets, etc. Nous proposons dans ce chapitre un nouvel algorithme itératif pour la reconstruction d'arrière-plan, où l'estimation en cours de l'arrière plan est utilisée pour évaluer quels pixels de l'image appartiennent à l'arrière-plan, et une nouvelle estimation de l'arrière plan est calculée en utilisant uniquement ces pixels. Nous montrons alors que l'algorithme proposé, qui utilise la descente de gradient stochastique pour ses propriétés de régularisation, est plus précis que l'état de l'art sur l'exigeant benchmark SBMnet, en particulier pour les courtes séquences vidéo avec un faible fréquence d'images, et est aussi rapide, atteignant une moyenne de 52 images par seconde sur ce jeu de données lorsqu'il est paramétré pour une précision maximale et en utilisant une carte graphique et une implémentation en Python.

3.2 Abstract

The goal of background reconstruction, also called background generation, is to recover the background image of a scene from a sequence of frames showing this scene cluttered by various moving objects. This task is fundamental in image analysis, and is generally the first step before more advanced processing, but difficult because there is no formal definition of what should be considered as background or foreground and the results may be severely impacted by vari-

ous challenges such as illumination changes, intermittent object motions, highly cluttered scenes, etc. We propose in this chapter a new iterative algorithm for background reconstruction, where the current estimate of the background is used to guess which image pixels are background pixels and a new background estimation is performed using those pixels only. We then show that the proposed algorithm, which uses stochastic gradient descent for improved regularization, is more accurate than the state of the art on the challenging SBMnet dataset, especially for short videos with low frame rates, and is also fast, reaching an average of 52 fps on this dataset when parameterized for maximal accuracy using acceleration with a graphics processing unit (GPU) and a Python implementation.

3.3 Introduction

We consider in this chapter the task of static background reconstruction: starting from a sequence of images $\mathcal{X} = X_1, ..., X_N$ of a scene showing moving objects, for example cars, bikes or pedestrians, the goal is to recover the image of the background of this scene, without any of the moving objects. This task is fundamental in image analysis: The moving objects appearing in the scene may be considered as a nuisance, and background reconstruction allows to remove them completely and focus on the analysis of the background, for example to localize or map the scene. More frequently, for example for video surveillance or traffic monitoring, the moving objects are the main object of interest and the background itself is considered as a nuisance, so that background reconstruction is a first step which can be used to extract and analyze the moving objects of the scene. The task of background reconstruction should not be confused with the task of background modeling, which involves building a statistical model of the background images whereas the task of background reconstruction requires to predict a unique background image.

It is often assumed that all the images $X_1, ..., X_N$ share the same background, which is then called a static background. In this case, the output of the algorithm is composed of only one background image $\hat{X}$. It is however also possible that the backgrounds are slightly different in each image, for example if the illumination conditions change or if the camera is moving. In this situation, which will be handled in chapter 4 we expect a background reconstruction algorithm to output a sequence of backgrounds $\hat{X}_1, .., \hat{X}_N$ and we say that the background reconstruction is dynamic. In this chapter, we consider the problem of static background reconstruction.

This problem is a difficult because there is no formal definition of what should be considered as background or foreground. Moving trees, fountains and moving shadows are examples of instances that are usually considered as belonging to the background although they show moving features. Other challenges such as illumination changes or the presence of objects staying still for a short time (a problem called intermittent motion) may severely impact the quality of a background reconstruction model.

One should distinguish between online methods, where the length of the dataset is unknown and the background reconstruction algorithm has to update the background model in real-time and batch methods, where the algorithm is provided with a fixed dataset. The method proposed in this chapter is a batch

method.

The main contribution of this chapter are the followings:

- We implement a new consistency criterion for background estimation: The background estimate produced by a background estimation method should not change if we perform the background estimation using only pixels that are considered as background pixels with regards to this background estimate.
- We then show that this consistency criterion can be described as an optimization criterion and that the associated optimization problem can be efficiently solved using stochastic gradient descent.

The chapter is organized as follows: In Section 3.4, we review related work in static background reconstruction. In Section 3.5, we describe the proposed algorithm. Experimental results are then provided in Section 3.6.

3.4 Related Work

Temporal median filtering (TMF) [Piccardi, 2004] simply computes the background color for a pixel p as the median of the colors of this pixel on all the images $X_1, .., X_N$. Despite its simplicity, this algorithm and its variant temporal median filter with Gaussian filtering (TMFG) [Liu et al., 2016] perform very well on several scene categories.

The current state-of-the-art models for unsupervised fixed background reconstruction are the Superpixel motion detection algorithm (SPMD) [Xu et al., 2019b] and LaBGen-OF [Laugraud and Van Droogenbroeck, 2017]. Both of these models, as well as the frame selection method and efficient background estimation procedure (FSBE) [Djerida et al., 2019], implement the idea that the regions of the input frames showing foreground objects should not be considered to compute the background.

SPMD first selects the longest sequence with stable illumination, then uses superpixel segmentation [Achanta et al., 2012], and removes all superpixels with contain at least one moving pixel using a frame difference method to detect moving pixels. The various pixel values associated with one pixel position are then clustered, and the median value of the best cluster is selected to produce the background value. Removing superpixels associated with moving pixels for background initialization is also developed in [Zhou et al., 2020].

LaBGen and LaBGen-P [Laugraud et al., 2017, Laugraud et al., 2016] assume that a background/foreground segmentation algorithm is available. For a given pixel or spatial patch, these models select the frames showing the lowest number of foreground pixels, and then perform a pixel-wise median filtering. LaBGen-OF is a variant which uses an optical flow algorithm [Laugraud and Van Droogenbroeck, 2017]. LaBGen-semantic is another variant with uses a supervised semantic segmentation model [Laugraud et al., 2018]. This model has also been adapted [Yu and Guo, 2019] to detect illumination changes and use only a subsequence with stable illumination conditions.

The FSBE algorithm (frame selection and background estimation) [Djerida et al., 2019] assumes that an optical flow algorithm is available. It first selects a sequence of frames where the illumination conditions do not change too much.

Using the optical flow algorithm, it classifies as background all pixels which have an optical flow magnitude below some threshold and corrects this classification if it detects high dynamic motion or foreground intermittent motion in the sequence. It then takes the pixel-wise average of the selected background pixels.

Instead of only removing pixels or patches which show moving objects before performing temporal median filtering, some models [Cohen, 2005,Xu and Huang, 2008] try to select for each pixel only one patch from the various frames, which is considered to be the best candidate to represent the background, so that temporal median filtering is not needed. Photomontage [Agarwala et al., 2004] builds the background as a seamless montage composed of patches extracted from the images $X_1, .., X_N$ so that the likelihood of the color at each pixel is maximum with respect to the probability distribution function formed from the color histogram of all pixels in the span.

Some models try to benefit from the fact that if the content of the background is known in some part of the image, it is easier to distinguish between background and foreground objects in adjacent parts of the image, using a spatial or temporal consistency criterion. The neighborhood exploration based background initialization (NExBI) algorithm [Mseddi et al., 2019] divides the frame in blocks, and perform a temporal clustering for each block location. A preliminary partial background model is then created for the blocks that remain stable during all the sequence, i.e., where all the image patches associated with this block form only one cluster, and then iteratively extended to the whole image as a puzzle game by enforcing consistency between candidate background blocks and the partial background model. Other iterative block completion models have been proposed in [Baltieri et al., 2010, Colombari and Fusiello, 2010, Hsiao and Leou, 2013, Lin et al., 2009, Ortego et al., 2016, Sanderson et al., 2011].

Another approach which has been investigated for background reconstruction is to consider the sequence $\mathcal{X} = X_1, ..., X_n$ as a 3D tensor or a spatiotemporal matrix and to decompose it as the sum of a low-rank part, which is assumed to be representative of the background, and a sparse part, which should be representative of the foreground objects. For example, the Motion-assisted spatiotemporal clustering of low-rank algorithm (MSCL) [Javed et al., 2017], which is a dynamic background reconstruction model using robust principal component analysis (RPCA) [Candès et al., 2011], is able to obtain better results than state of the art fixed background reconstruction models on the scene background modeling (SBMnet) dataset using this method, although it is not directly comparable to those models because it requires some human supervision to select the final frame $\hat{X}$ from the various predicted backgrounds $\hat{X}_1, ..., \hat{X}_n$ associated with the frames $X_1, ..., X_n$. Another linear method proposed to extract a low rank background is to apply singular value decomposition (SVD) to spatiotemporal slices of the tensor $\mathcal{X}$, consider that the first principal subspace is associated with the background, and use the other components to detect foreground objects, which can then be excluded from the background computation [Kajo et al., 2018, Kajo et al., 2020].

The background estimation by weightless neural network (BEWIS) [De Gregorio and Giordano, 2015] and self-organizing background subtraction (SOBS) algorithms [Maddalena and Petrosino, 2008a, Maddalena and Petrosino, 2016, Maddalena and Petrosino, 2012] involve weightless neural networks, which are used as containers to build a statistical model of the background.

The current top performing algorithms for background reconstruction do not use deep learning techniques, but several papers have proposed to use them for fixed background reconstruction:

Fully-concatenated Flownet (FC-Flownet) [Halfaoui et al., 2016] is a convolutional network with an architecture similar to a U-net which is used to predict a background from a set of 20 color images in a single inference step. Due to memory restrictions, the images are cut in superposed 64x64 patches, and the 20 patches associated with one location are given as input to the convolutional network. The output patches are then aggregated to build the background. The network is trained end-to-end using samples and ground truths coming from 54 different sequences.

Background modeling Unet (BM-Unet) [Tao et al., 2017] is a background reconstruction model which also uses a U-net network but is trained without any supervision or ground-truth data and can perform both fixed and dynamic background reconstruction. For fixed background reconstruction, it is trained with pairs of random images sampled from one frame sequence. Using the first image, the U-net network predicts a probability distribution over the possible 256 values of each pixel of the output image, and the second image is used as a target.

Deep context prediction (DCP) [Sultana et al., 2019] considers the background reconstruction problem as an inpainting problem: Using an optical flow algorithm, it first computes the motion mask associated with the current frame and removes from this frame the pixels associated with this motion mask. It then uses a multi-scale neural path synthesis network [Yang et al., 2017] to fill the holes in the image and obtain a clean background. Other data reconstruction methods using classical matrix completion or exemplar-based approaches are also possible [Colombari et al., 2005, Sobral et al., 2015, Sobral and hadi Zahzah, 2017].

We refer to available surveys [Bouwmans et al., 2017,Bouwmans et al., 2019] for a more detailed description of related work.

3.5 Proposed Algorithm for Background Reconstruction

3.5.1 Motivation

We have noted in the previous section the good results of temporal median filtering, despite its simplicity, and observe that the two best unsupervised algorithms for background reconstruction, SPMD and LabGen-OF, also use some form of temporal median filtering. One can intuitively understand that background reconstruction involves performing some form of averaging of the input frames, and that computing the median will give better results than computing the average of the frames because the median is more robust to outliers.

We note however that using median filtering on color images may lead to inconsistencies. Let us for example consider RGB images showing a red background with large green and blue foreground objects. Assume that in the sequence considered, each red background pixel is masked by a green object during 26% of sequence duration and by a blue object during another 26% of the sequence duration. The red color channel of any pixel will then be equal to zero

during 52% of the sequence, and the blue and green channels are also equal to zero during 74% of the sequence. As a consequence, the result of median filtering on such a sequence is a uniform black image, which is clearly not satisfactory.

One can think that a better method to select the background color of an image from a frame sequence would be first to guess in each frame which pixels are background pixels and then to consider only those pixels for temporal median filtering. However, to be able to guess which pixels are background pixels, we need to have some estimate of the background. The main idea introduced in this chapter is that we can successfully build an iterative optimization process for background reconstruction, using the current estimate of the background to guess which pixels are background pixels and then refining the estimate of the background by performing temporal median filtering on those pixels only.

3.5.2 Bootstrap weights

We observe [Huber, 1964] that temporal median filtering can be described as a minimization problem associated with a L_1 error loss. More precisely, for a sequence of color images $X_1, .., X_N$ of size $h \times w$, noting $x_{n,c,i,j}$ the value (normalized in the range $[0, 1]$) of the pixel associated with the image X_n and the color channel c at position (i, j) with $1 \leq i \leq h$ and $1 \leq j \leq w$, the L_1 error loss associated with some background reconstruction $\hat{X}$ can be described as

$$\mathcal{L}_1(\hat{X}, (X_n)_{1 \leq n \leq N}) = \frac{1}{N} \sum_{n=1}^{N} L_1(\hat{X}, X_n) \tag{3.1}$$

with

$$L_1(\hat{X}, X_n) = \frac{1}{hw} \sum_{i=1,j=1}^{h,w} \sum_{c=1}^{3} |\hat{x}_{c,i,j} - x_{n,c,i,j}|, \tag{3.2}$$

and it is immediate that if we take each $\hat{x}_{c,i,j}$ to be a median of the sequence $(x_{n,c,i,j})_{1 \leq n \leq N}$, then we obtain a minimum of this loss function, considering that the derivative of $|\hat{x}_{c,i,j} - x_{n,c,i,j}|$ with respect to $\hat{x}_{c,i,j}$ is equal to 1 if $\hat{x}_{c,i,j} - x_{n,c,i,j} > 0$ and -1 if $\hat{x}_{c,i,j} - x_{n,c,i,j} < 0$.

We now consider the foreground pixels as outliers and propose to bootstrap the current estimate of the background to smoothly restrict this loss function to background pixels and reduce the influence of the foreground pixels. We then give a low weight, called a bootstrap weight, to the pixel-wise error terms $\sum_{c=1}^{3} |\hat{x}_{c,i,j} - x_{n,c,i,j}|$ associated with foreground pixels in the loss function. These bootstrap weights are computed in the following way (Figure 3.1):

Let us note $l_{n,i,j}$ the sum of the L_1 errors for each color at the pixel (i, j) between the predicted image $\hat{X}$ and the input image X_n for all the color channels:

$$l_{n,i,j} = \sum_{c=1}^{3} |\hat{x}_{c,i,j} - x_{n,c,i,j}| \tag{3.3}$$

If at least one of the color channels give a high error, then $l_{n,i,j}$ is large and we will consider that the pixel (i, j) of the image X_n is a foreground pixel. We then build a soft foreground mask $m_n \in [0, 1]^{h \times w}$ for the image X_n using the formula

$$m_{n,i,j} = \tanh\left(\frac{l_{n,i,j}}{\tau_1}\right) \tag{3.4}$$

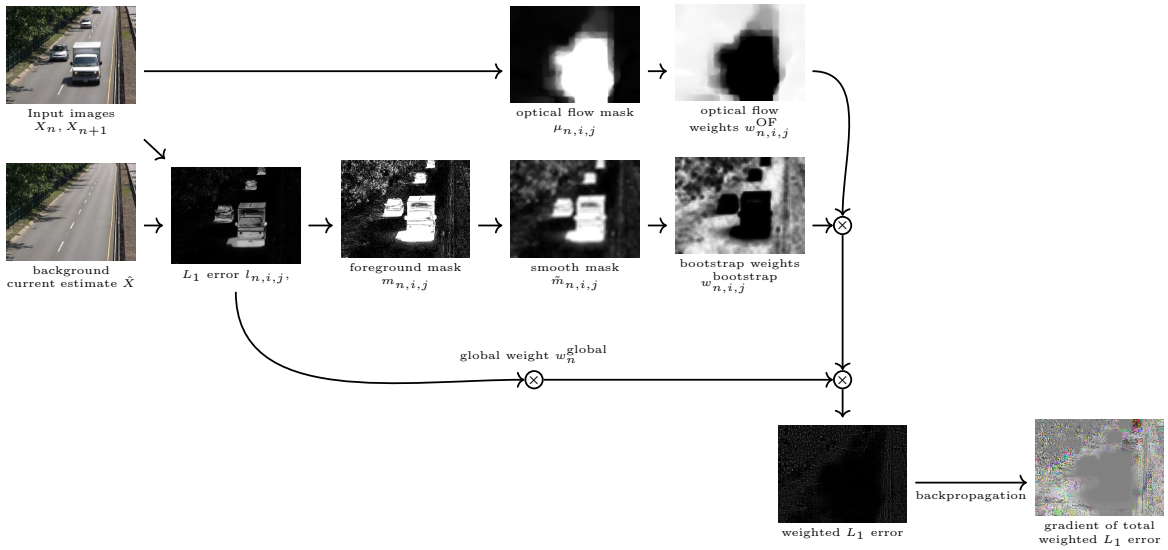


Figure 3.1: Schematic of loss function and gradient computation (Images are normalized in the range [0,1]).

where τ_1 is some positive hyperparameter, which can be considered as a soft threshold. As a consequence, $m_{n,i,j}$ is close to zero for values of $l_{n,i,j}$ close to zero (background pixels), and close to 1 for values of $l_{n,i,j}$ which are significantly larger than τ_1 (foreground pixels).

This mask will however be noisy. We then compute a spatially smoothed version $\tilde{m}_{n,i,j}$ of this mask by averaging using a square kernel of size $(2k+1) \times (2k+1)$, with $k = \lfloor w/r \rfloor$ (where w is the image width and r is some integer hyperparameter):

$$\tilde{m}_{n,i,j}(\hat{X}, X_n) = \frac{1}{(2k+1)^2} \sum_{l=-k,p=-k}^{l=k,p=k} m_{n,i+l,j+p} \tag{3.5}$$

We then finally define the associated pixel-wise bootstrap weights $w^{\mathrm{bootstrap}}_{n,i,j}$ as

$$w^{\mathrm{bootstrap}}_{n,i,j} = e^{-\beta \tilde{m}_{n,i,j}}, \tag{3.6}$$

where β is some positive hyperparameter, which we call the bootstrap coefficient.

For pixels which are considered to be background pixels ($\tilde{m}_{n,i,j} \simeq 0$), this weight will be close to 1 and will not change the pixel-wise loss terms $\sum_{c=1}^{3} |\hat{x}_{c,i,j} - x_{n,c,i,j}|$ associated with these pixels. However for pixels which are considered as foreground pixels ($\tilde{m}_{n,i,j} \simeq 1$), this weight will have a very low value close to $e^{-\beta}$, which means that the associated pixel-wise loss terms will get a very low importance in the loss function.

3.5.3 Optical Flow Weights

We have seen that background reconstruction algorithms could be improved by using informations provided by optical flow models to remove parts of an image

showing moving objects. We use the same approach to improve the loss function $\mathcal{L}_1$. We then define optical flow weights associated with each pixel $x_{n,i,j}$ which will be close to zero if this pixel appears to be a moving pixel and has to be removed from the loss function computation. These weights are computed in the following way (cf. Figure 3.1):

We use an external algorithm (OpenCV implementation of Dense Inverse Search algorithm [Kroeger et al., 2016]) to obtain an estimate of the magnitude $\phi_{n,i,j}$ of the optical flow associated with each pixel (i, j) of an image X_n. We chose this algorithm because it is very fast compared to other available optical flow implementations. We first normalize $\phi_{n,i,j}$ with respect to the image width w and then define an optical flow mask $\mu_{n,i,j}$ using the formula

$$\mu_{n,i,j} = \min\left(1, \frac{\phi_{n,i,j}}{w\tau_2}\right), \tag{3.7}$$

where the hyperparameter τ_2 can also be considered as a threshold. This mask is then equal to 1 for high values of the optical flow $\phi_{n,i,j}$, which suggests that the associated pixels show a moving object, and it is close to zero if no motion is detected by the optical flow algorithm at the associated pixel.

The weight associated with this optical flow mask is then defined as

$$w^{\mathrm{OF}}_{n,i,j} = e^{-\phi\mu_{n,i,j}}, \tag{3.8}$$

where ϕ is another positive hyperparameter. This weight will then be equal to 1 if no motion is detected at the associated pixel, and close to $e^{-\phi}$ if a significant motion is detected, which suggests that the associated pixel is not a background pixel. $w^{\mathrm{OF}}_{n,i,j}$ is, however, set to 1 for all pixels on short videos (less than ten images), considering that optical flows computed from sequences with very low frame rates are not reliable.

3.5.4 Abnormal Image Weights

If the number of images in the dataset is large, we can afford to give a low weight to images which appear to be abnormal and can be considered as outliers, for example if the illumination conditions are different on these images compared to the predicted background, or if there are too many pixel errors on the image. We then first compute the average L_1 error $\bar{l}_n$ of the image X_n as

$$\bar{l}_n = \frac{1}{hw}\sum_{i,j} l_{n,i,j} \tag{3.9}$$

and define a global weight associated with each image X_n as

$$w^{\mathrm{global}}_n = e^{-\gamma\bar{l}_n}, \tag{3.10}$$

where γ is another positive hyperparameter. As a consequence, this weight w^{global}_n will be close to zero if the image X_n is globally very different from the current estimate $\hat{X}$ of the background. We use this global weight if the size of the dataset is greater than 10. It should be noted that this weight is not pixel-specific, as opposed to the bootstrap weights and optical flow weights, but is assigned to a complete frame.

3.5.5 Management of Intermittent Motion

Existing benchmarks for background reconstruction require that objects which remain still for a long time in the sequence be considered as foreground objects if they are moving during some part of the sequence. This challenge is very difficult and is not addressed by the previous weights. In order to handle it, we follow Javed et al. [Javed et al., 2016, Javed et al., 2017] and remove from the frames sequence all frames which are not showing any motion. More precisely, we first compute the maximum μ_n^* of the optical flow mask values $\mu_{n,i,j}$ of the image X_n as defined in previous section, and remove this image if $\mu_n^* < \tau_3$, where τ_3 is another threshold hyperparameter. The motivation of this suppression is that it appears that images containing still foreground objects are often motionless images, so that removing them improves the robustness of the proposed model against the intermittent motion issue. We apply this motionless frame suppression when the number of frames in the sequence is higher than 10, considering as in previous section, that removing frames when the number of frames is very low will impact negatively the quality of the results. We note $N' \leq N$ the number of frames after motionless frame suppression.

3.5.6 Statement of the Optimization Problem

Finally, the loss function is adapted using these weights and becomes the following:

$$\mathcal{L}_W(\hat{X}, (X_n)_{1\leq n\leq N'}) = \frac{1}{N'hw} \sum_{n=1,i=1,j=1}^{N',h,w} w_n^{\text{global}} w_{n,i,j}^{\text{bootstrap}} w_{n,i,j}^{OF} \sum_{c=1}^{3} |\hat{x}_{c,i,j} - x_{n,c,i,j}|. \tag{3.11}$$

We are then interested to solve the following optimization problem: *Considering the dataset $(X_n)_{1\leq n\leq N'}$, find an image $\hat{X}$ so that, when the weights w_n^{global} and $w_{n,i,j}^{bootstrap}$ are considered as constants, the loss function $\mathcal{L}_W(\hat{X}, (X_n)_{1\leq n\leq N'})$ is minimal with respect to $\hat{X}$.*

We can find a solution to this problem by performing an iterative computation of the weighted median of the images using the various weights defined in the previous paragraph followed by an update of the weights, a process similar to the classical iterated reweighted least square algorithm [Lawson, 1961]. We observe, however, that the images produced using this method are not smooth and that additional regularization is necessary. We then propose to use stochastic gradient descent on the loss function $\mathcal{L}_W(\hat{X}, (X_n)_{1\leq n\leq N'})$ using standard deep learning tools. The pixel values $\hat{x}_{c,i,j}$ are then considered as parameters and optimized using stochastic gradient descent (Figure 3.2).

It should be noted that performing a stochastic gradient descent on this loss function is not equivalent to minimizing it: During the optimization process, the weights $w_{n,i,j}^{\text{boostrap}}$ and w_n^{global} depend on the current estimation of the background and change; we then call these weights dynamic weights. At each iteration they are, however, considered as fixed so that we do not compute and use the gradient of the loss function with respect to the value of these weights.

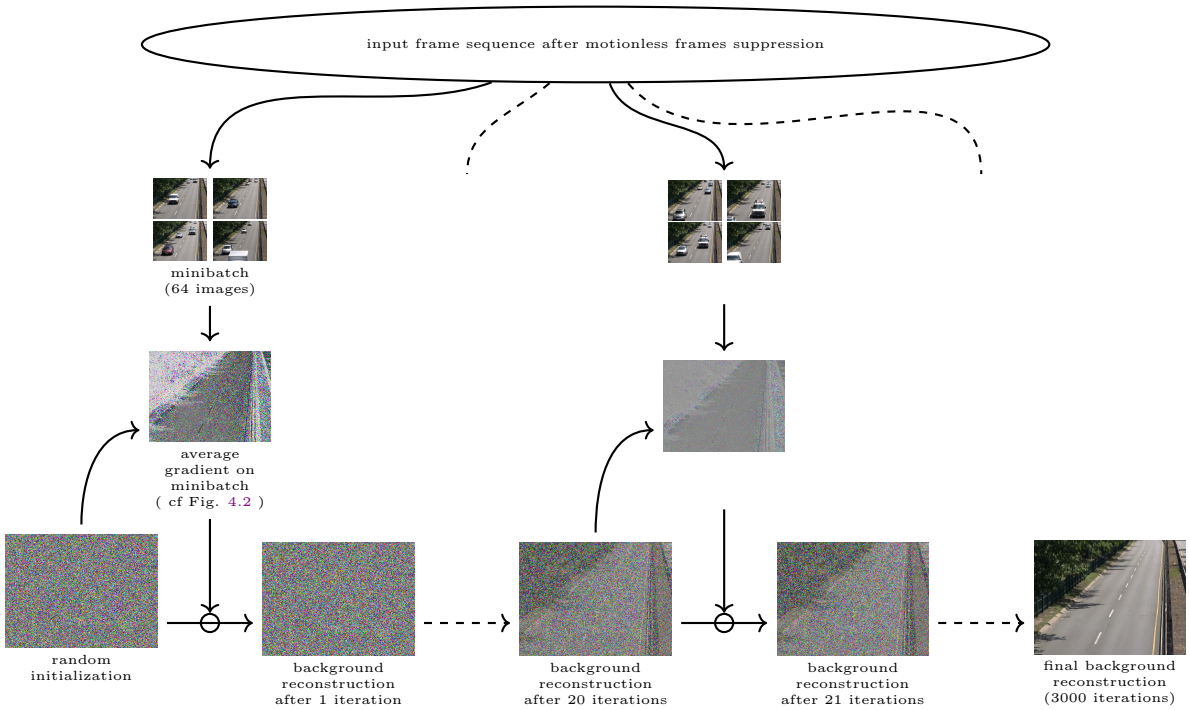


Figure 3.2: Overview of the stochastic gradient descent optimization process

3.6 Evaluation of the Proposed Model

Two public benchmarks are available for the evaluation of fixed background reconstruction models: the SBMnet dataset [Jodoin et al., 2017] and the SBI dataset [Maddalena and Petrosino, 2015]. We first provide a quantitative evaluation of the proposed model on those two datasets. We then perform an ablation study and some computation speed measurements.

3.6.1 Implementation Details

A desktop computer with an Intel Core i7 7700K@4,2GHz CPU and a Nvidia RTX 2080 TI GPU is used for this experiment.The model is implemented in Python using the Pytorch framework and is publicly available on the Github platform. We use the Adam optimizer [Kingma and Ba, 2015], with learning rate 0.03 and batch size 64, reduced by a factor of 10 when 3/4 of the epochs have been computed. The number of epochs depends on the size of the dataset and is adjusted so that the total number of optimization iterations is close to 3000, with a minimum of two epochs. In order to accelerate computations, each frame sequence is fully loaded in the GPU video RAM during the optimization process. A manual hyperparameter search has been performed using the video sequences of the SBI and SBM datasets for which a ground truth is available. The hyperparameters have then been set to the following values: $\beta = 6, \phi = 2, \gamma = 3, r = 75, \tau_1 = 0.25, \tau_2 = 255/40000, \tau_3 = 240/255$. Before starting the optimization, background image pixel color values are initialized with random

numbers sampled from a uniform distribution between 0 and 1. The DIS optical flow OpenCV implementation is used with the FAST preset mode. In order to obtain a low gradient when $l_{n,i,j}$ is close to zero, we replace the expression $|\hat{x}_{c,i,j} - x_{n,c,i,j}|$ with a smooth L_1 loss using a threshold equal to 3 (assuming the pixel values are scaled in the range 0-255). When $|\hat{x}_{c,i,j} - x_{n,c,i,j}|$ is lower than 3, we replace it with the quadratic expression $0.5(\hat{x}_{c,i,j} - x_{n,c,i,j})^2/3$, otherwise we replace it with $|\hat{x}_{c,i,j} - x_{n,c,i,j}| - 0.5 \times 3$.

3.6.2 Evaluation on SBMnet dataset

The SBMnet dataset [Jodoin et al., 2017] (http://scenebackgroundmodeling.net) is composed of 79 sequences, which have been selected to cover a wide range of challenges and are representative of typical indoor and outdoor visual data captured today in surveillance, smart environment, and video database scenarios. The dataset includes the following eight categories with associated challenges: basic, intermittent motion, clutter, jitter, illumination changes, background motion, very long and very short. Although this dataset is freely available on the SBMnet website, ground truth images are publicly available for only 18 frame sequences, either on the SBMnet website or on the SBI dataset website. In order to benchmark a new algorithm, one has to submit the predicted fixed background images associated with each frame sequence to the website, which performs the evaluation of the submitted results.

Six criteria are computed to evaluate the accuracy of background reconstruction:

- Average Gray-level Error (AGE);
- Percentage of Error Pixels (pEPs);
- Percentage of Clustered Error Pixels (pCEPs);
- Multi-Scale Structural Similarity Index (MS-SSIM);
- Peak-Signal-to-Noise-Ratio (PSNR);
- Color image Quality Measure (CQM).

We refer to [Jodoin et al., 2017] for the full definition of these criteria. A good background reconstruction should minimize the criteria AGE, pEPs and pCEPs, but maximize the criteria MS-SSIM, PSNR and CQM. We have computed the 79 background images using the proposed algorithm and uploaded the reconstructed backgrounds to the SBMnet website, which provided the evaluation results described in Tables 3.1 and 3.2.

We provide a comparison of the proposed model with models that are fully unsupervised, i.e., which do not use a supervised segmentation model (such as LabGen-semantic) and do not require any human supervision. The proposed model, named BB-SGD (background bootstrapping using stochastic gradient descent) obtains a better average score than all referenced unsupervised models on all criteria as shown in Table 3.1. Table 3.2 lists AGE results per category of the SBMnet dataset. It shows that the proposed models shows better AGE results than all referenced unsupervised models on 4 categories: basic, clutter, background motion and very short video, with a 15% accuracy improvement

Table 3.1: Evaluation results per criteria on the SBMnet 2016 dataset. ↓ indicates lower score is better, ↑ indicates higher score is better. Source: SBMnet website http://pione.dinf.usherbrooke.ca/results/294/ accessed on 20 November 2021. Available in 2022 on the following link: https://web.archive.org/web/20220618062451/http://pione.dinf.usherbrooke.ca/results/294/
.

Method	Average AGE ↓	Average pEPs↓	Average pCPEPs↓	Average MS-SSIM↑	Average PSNR↑	Average CQM↑
BB-SGD (ours)	**5.6266**	**0.0447**	**0.0147**	**0.9478**	**30.4016**	**31.2420**
SPMD [Xu et al., 2019b]	6.0985	0.0487	0.0154	0.9412	29.8439	30.6499
LabGen-OF [Laugraud and Van Droogenbroeck, 2017]	6.1897	0.0566	0.0232	0.9412	29.8957	30.7006
FSBE [Djerida et al., 2019]	6.6204	0.0605	0.0217	0.9373	29.3378	30.1777
BEWIS [De Gregorio and Giordano, 2015]	6.7094	0.0592	0.0266	0.9282	28.7728	29.6342
NExBI [Mseddi et al., 2019]	6.7778	0.0671	0.0227	0.9196	27.9944	28.8810
Photomontage [Agarwala et al., 2004]	7.1950	0.0686	0.0257	0.9189	28.0113	28.8719
SOBS [Maddalena and Petrosino, 2016]	7.5183	0.0711	0.0242	0.9160	27.6533	28.5601
Temporal Median Filter [Piccardi, 2004]	8.2761	0.0984	0.0546	0.9130	27.5364	28.4434

Table 3.2: Evaluation results for the AGE criterion per category on the SBMnet 2016 dataset. Source: SBMnet website http://pione.dinf.usherbrooke.ca/results/294/ accessed on 20 November 2021.

Method	Basic	Interm. Motion	Clutter	Jitter	Illumin. Changes	Backgr. Motion	Very Long	Very Short
BB-SGD (ours)	**3.7881**	4.8898	**3.8776**	9.5374	4.5227	**8.5607**	5.6494	**4.1872**
SPMD [Xu et al., 2019b]	3.8141	**4.1840**	4.5998	9.8095	**4.4750**	9.9115	6.0926	5.9017
LabGen-OF [Laugraud and Van Droogenbroeck, 2017]	3.8421	4.6433	4.1821	9.2410	8.2200	10.0698	4.2856	5.0338
FSBE [Djerida et al., 2019]	3.8960	5.3438	4.7660	10.3878	5.5089	10.5862	6.9832	5.4912
BEWIS [De Gregorio and Giordano, 2015]	4.0673	4.7798	10.6714	9.4156	5.9048	9.6776	**3.9652**	5.1937
Photomontage [Agarwala et al., 2004]	4.4856	7.1460	6.8195	10.1272	5.2668	12.0930	6.6446	4.9770
SOBS [Maddalena and Petrosino, 2016]	4.3598	6.2583	7.0590	10.0232	10.3591	10.7280	6.0638	5.2953
Temporal Median Filter [Piccardi, 2004]	3.8269	6.8003	12.5316	**9.0892**	12.2205	9.6479	6.9588	5.1336

on the very short video category compared to the best unsupervised model in this category, which illustrates the efficiency of the bootstrapping mechanism introduced in the proposed model considering that for these sequences, the optical flow weights and global weights are not used and no frame is suppressed.

3.6.3 Evaluation on SBI Dataset

The SBI dataset [Maddalena and Petrosino, 2015] (https://sbmi2015.na.icar.cnr.it/SBIdataset.html accessed on 20 November 2021) is composed of 14 image sequences. Ground truth backgrounds are available for all sequences. We use the Matlab tool available on the SBI website for fair comparison with other models, but do not report the CQM results considering that other sequences were evaluated with a Matlab tool which included a bug for the CQM computation, as indicated in the SBI website. We run the proposed model on the SBI dataset using the same hyperparameters as those used for the SBMnet dataset. The results of this evaluation are listed in Table 3.3 and show that the proposed model obtains better results than all other compared unsupervised models for the evaluation criteria AGE, MS-SSIM and PSNR, and is ranked second for the criteria pEPs and pCEPs.

Table 3.3: Evaluation results per criteria on the SBI dataset. ↓ indicates lower score is better, ↑ indicates higher score is better.

Method	Average AGE ↓	Average pEPs ↓	Average pCEPs ↓	Average MS-SSIM ↑	Average PSNR ↑
BB-SGD (ours)	**2.4644**	0.0083	0.0058	**0.9896**	**37.6227**
LabGen-OF [Laugraud and Van Droogenbroeck, 2017]	2.7191	0.0145	0.0106	0.9824	35.9758
SS-SVD [Kajo et al., 2018]	2.7479	0.0345	0.0907	0.9464	31.8116
LabGen [Laugraud et al., 2017]	2.9945	0.0139	0.0092	0.9764	35.2028
NExBI [Mseddi et al., 2019]	3.0547	**0.0077**	**0.0027**	0.9835	35.3078
BEWIS [De Gregorio and Giordano, 2015]	3.8665	0.0242	0.0142	0.9675	32.0143
Photomontage [Agarwala et al., 2004]	5.8238	0.0469	0.0372	0.9334	31.8573
SOBS [Maddalena and Petrosino, 2016]	3.5023	0.0415	0.0222	0.9765	35.2723
Temporal Median Filter [Piccardi, 2004]	10.3744	0.1340	0.1055	0.8533	28.0044

3.6.4 Ablation Study

In order to check the contribution of the various weights described in this chapter, we provide results obtained using truncated versions of the proposed model while keeping the hyperparameters fixed: Version 0 does not use any weight and does not remove motionless frames, and is then equivalent to temporal median filtering. Version 1 uses only the optical flow weights and does not remove motionless frames. Version 2 uses both optical flow weights and global weights and does not remove motionless frames. Version 3 uses bootsrap weights, global weights and optical flow weights, but does not remove motionless frames.

The AGE scores obtained by these truncated models on the 18 videos of the SBMnet dataset for which a ground truth is available and using the evaluation tool available on the SBMnet website are provided in Table 4.7. They show that temporal median filtering (v0) gives the best results for five scenes, confirming that this is a good baseline. Introducing optical flow weights (v1) improves average AGE scores on scenes of the "clutter" category, but has no beneficial impact on other categories. Adding global weights (v2) has a positive impact on the "illumination change" category, which was expected, but also on the "clutter" category". Adding bootstrap weights has an impact on the "clutter" category, but also on the "short video" category. Finally, removing motionless frames, which leads to the full model, has a positive impact on the "intermittent motion" category, which was expected, but also on the scene "boulevardJam" of the "clutter" category, which also shows some intermittent motions.

3.6.5 Computation Time

We have performed computation times measurements and tested the impact of reducing the number of optimization iterations, while keeping all other parameters frozen, excluding the learning rate. The results of these experiments are provided in Table 3.5. The total computation times necessary to reconstruct the 79 backgrounds from the associated video sequences of the SBMnet dataset are estimated by performing a sequential computation for all the videos, so that the computation times indicated in this table are the sum of the computation times of each of the 79 videos. If we divide the number of frames of the full dataset (73,355) with the total computation time of the proposed model, which is 1409 s, we obtain an average of 52 frames per second (fps). Table 3.5 shows, however, that the number of optimization iterations can be reduced from 3000 to 250,

Table 3.4: AGE scores obtained using various truncated versions of the algorithm on 18 SBMnet sequences where a ground truth background is available.

Category	**Video**	**Truncated Model Version**				**Full Model**
		v0	**v1**	**v2**	**v3**	
background motion						
	advertisementBoard	1.61	1.62	1.60	1.34	1.71
basic						
	511	3.42	3.44	3.43	3.44	3.43
	Blurred	1.80	1.69	1.68	1.68	1.61
clutter						
	Foliage	32.87	5.86	3.62	3.41	3.37
	Board	21.37	6.78	7.84	7.37	7.39
	People and Foliage	31.36	9.66	3.75	2.54	2.60
	boulevardJam	21.37	15.89	19.5	11.0	2.03
illumination change						
	CameraParameter	11.49	22.19	2.16	2.81	2.95
intermittent motion						
	busStation	5.31	5.40	5.47	5.67	5.32
	Candela_m1.10	4.93	5.09	5.18	5.21	2.81
	CaVignal	12.57	12.61	13.58	14.04	2.05
	AVSS2007	10.98	10.32	10.25	10.01	8.73
jitter						
	badminton	2.62	2.00	1.93	1.74	1.84
	boulevard	9.61	10.09	10.29	10.51	9.71
very long						
	BusStopMorning	3.68	3.66	3.64	3.62	3.61
very short						
	Toscana	8.79	8.80	8.79	3.30	3.30
	DynamicBackground	6.96	6.96	6.96	8.20	8.18
	CUHK_Square	2.77	2.77	2.77	2.99	2.98
Average AGE by category		8.06	7.53	4.94	4.51	3.75

increasing the average speed to 187 fps, without major impact on the overall accuracy of the algorithm. The computation times with such a low number of iterations are mainly associated with optical flow computations and JPEG images decoding.

Although the proposed model requires a GPU, these computation time measurements compare very favorably with the processing speeds reported by the authors of other models. The average computation speed of LabGen-OF is estimated to 5fps in [Laugraud and Van Droogenbroeck, 2017]. The computation speed of SPMD is estimated in [Xu et al., 2019b] to 1.6 fps for 640×480 images and 22.8 fps for 200×144 images using a Intel Core i7 2600@3.4Ghz CPU.

The asymptotic time complexity of the proposed algorithm is $\mathcal{O}(p^2)$ where $p = hw$ is the number of pixels of an image. It does not depend on the number of frames of the input frame sequence since a maximum of 64×3000 images are sampled from the input sequence (3000 minibatches of 64 images) and optical flow computations can be restricted to those images only. The quadratic expression $\mathcal{O}(p^2)$ is a consequence of Equation (3.5), which involves a kernel which has a size proportional to the size of the images for r fixed.

Table 3.5: Impact of reducing the number of iterations on average AGE score and computation time.

Number of Iterations **Learning Rate**	**100** **0.06**	**250** **0.03**	**500** **0.03**	**1000** **0.03**	**3000** **0.03**
Computation time for 79 videos of the SBMnet dataset (seconds)	337	391	482	666	1409
Average AGE by category on 18 videos of the SBMnet dataset listed in Table 4.7	4.07	3.83	3.80	3.76	3.75
Average AGE on SBI dataset	2.78	2.56	2.53	2.49	2.46

3.6.6 Image Samples

Figures 3.3 and 3.4 show for qualitative evaluation some examples of background reconstruction for sequences of the SBMnet dataset, with the associated ground-truth when it is available and a comparison with the results obtained with LabGen-OF and SPMD. The bottom five rows of Figure 3.4 show some examples of poor quality reconstructions suffering from challenging issues such as intermittent motion, headlights and moving trees.

3.6.7 Hyperparameter Tuning

The proposed model involves a significant number of hyperparameters. Although the default hyperparameters proposed in Section 3.6 allow us to obtain state-of-the-art performances on existing benchmarks, these hyperparameters may be fine-tuned to improve results on specific situations or use cases. We provide below some indications on the influence of the main hyperparameters:

- τ_1: the soft threshold used for computing soft foreground mask should be decreased for frame sequences with very low average illumination.
- τ_2: the soft threshold used for computing optical flow masks should be decreased for video sequences with high frame rates and increased for sequences with low frame rates, considering that optical flow values are lower for a high frame rate sequences and higher for a low frame rate sequences.
- Optical flow weight ϕ: as shown in the ablation study, the use of optical flow weights is only necessary for highly occluded scenes. More precise results may be obtained by setting this parameter to lower value if a high level of occlusion is not expected.
- r: the value of r is associated with the expected sizes of the foreground objects: If it is forecast that the scenes will contain only small foreground objects, this value may be increased on high definition images for faster training.
- Bootstrap coefficient β: a lower value of β leads to faster training, but decreases the ability to handle occlusions. A higher value of β may lead to slower or unstable training and artifacts in the final image.

- Global weight γ: increasing the value of γ may be useful to handle low intensity illumination changes.

3.7 Conclusion of chapter 3

We have presented in this chapter a new algorithm for fixed background reconstruction using stochastic gradient descent which is simple, fast using a GPU and is more accurate than the current state of the art.

The background reconstruction algorithm which has been developed in this chapter is however not sufficient for our purpose. It is quite rare that videos or image datasets show a fixed background. The backgrounds of any outdoor scene will for example be heavily affected by the illumination changes caused by sun movements and weather variations. In order to build an accurate background reconstruction from a video, we will then study in the next chapter the task of dynamic background reconstruction, which requires the reconstruction of a different background for each input image.

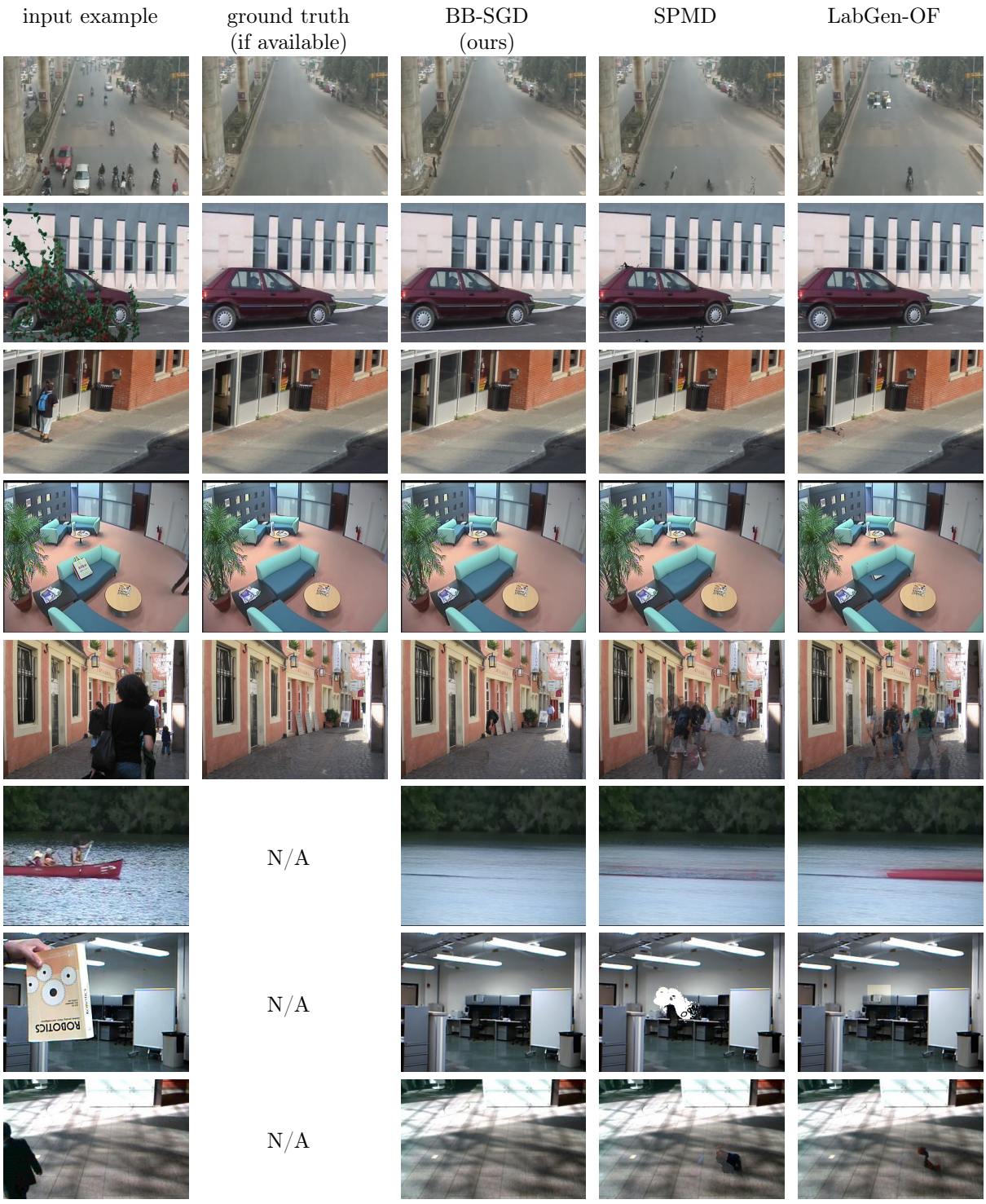


Figure 3.3: Examples of background reconstruction using the proposed model and comparison with SPMD and LabGen-OF.

Figure 3.4: Examples of background reconstruction. The bottom five rows show examples of low quality reconstructions.

Chapter 4

Dynamic background reconstruction

4.1 Résumé en français

Malgré plusieurs décennies de recherche, la reconstruction d'arrière-plan dynamique et la segmentation des objets constituant l'avant-plan sont toujours considérées comme des problèmes ouverts à cause de difficultés variées telles que les variations de luminosité des images, les mouvements de caméra, ou le bruit d'arrière-plan causé par la turbulence de l'air ou les mouvements des arbres. Nous proposons dans ce chapitre de modéliser l'arrière-plan d'une séquence d'images comme une variété de petite dimension en utilisant un auto-encodeur, et comparons l'arrière-plan généré par l'auto-encodeur avec l'image originale afin de calculer le masque de segmentation arrière-plan/avant-plan. La principale nouveauté du modèle proposé est que l'auto-encodeur est aussi entraîné à prédire le bruit de l'arrière-plan, ce qui permet de calculer pour chaque image un seuil spécifique pour chaque pixel afin d'effectuer la segmentation d'avant-plan. Bien que le modèle proposé n'utilise aucune information temporelle ou relative aux mouvements des objets, elle dépasse l'état de l'art en matière de segmentation d'arrière-plan sur les benchmarks CDnet 2014 et LASIESTA, avec une avance significative sur les vidéos ou la caméra est en mouvement. Elle est aussi en mesure de réaliser de la reconstruction d'arrière-plan sur des jeux de données d'images qui ne constituent pas des séquences vidéos.

4.2 Abstract

Even after decades of research, dynamic scene background reconstruction and foreground object segmentation are still considered as open problems due various challenges such as illumination changes, camera movements, or background noise caused by air turbulence or moving trees. We propose in this chapter to model the background of a frame sequence as a low dimensional manifold using an autoencoder and compare the reconstructed background provided by this autoencoder with the original image to compute the foreground/background segmentation masks. The main novelty of the proposed model is that the

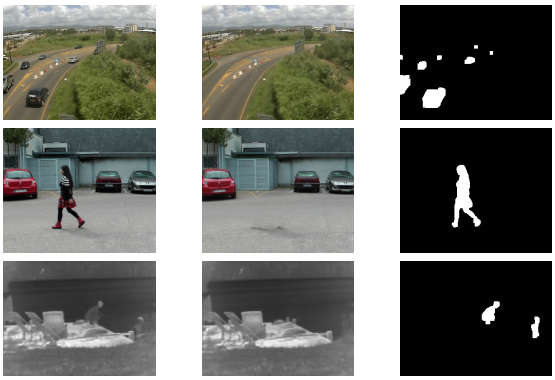


Figure 4.1: The proposed model takes as input a frame from the associated video (left column) and provides a reconstruction of the background (middle column) and a foreground mask (right column).

autoencoder is also trained to predict the background noise, which allows to compute for each frame a pixel-dependent threshold to perform the foreground segmentation. Although the proposed model does not use any temporal or motion information, it exceeds the state of the art for unsupervised background subtraction on the CDnet 2014 and LASIESTA datasets, with a significant improvement on videos where the camera is moving. It is also able to perform background reconstruction on some non-video image datasets.

4.3 Introduction

We consider in this chapter the tasks of dynamic background reconstruction and foreground/background segmentation, which can be described in the following way: The input is a sequence $\mathcal{X}$ of consecutive frames $X_1, .., X_N$ showing a scene cluttered by various moving objects, such as cars or pedestrians, and the expected output is a sequence $\hat{\mathcal{X}} = \hat{X}_1, .., \hat{X}_N$ of frames showing the backgrounds of each scene without those objects.

The foreground/background segmentation task similarly takes as input the same kind of frames sequence $X_1, .., X_N$, but the expected output is a sequence $\mathcal{M}$ of foreground masks $M_1, .., M_N$ whose values at the pixel p are equal to zero if this pixel shows the background in the considered frame, and equal to 1 if the background is masked by a foreground moving object at this pixel (Fig. 4.1). This task is often called background subtraction because the pointwise multiplication of the mask M_k and the input image X_k gives an image showing only the foreground moving objects present in X_k, the input image background being replaced by a black background.

Background substraction is a fundamental tool in image analysis and has been studied for more than 30 years [Wren et al., 1997], but is still considered an open problem due to the various challenges appearing in real applications: illumination changes, high level of occlusion of the background, background motions caused by moving trees or water, challenging weather conditions, presence of shadows, etc. The applications of background subtraction are very diverse [Garcia-Garcia et al., 2020]: road, airport, store, maritime

or military surveillance, observation of animals and insects, motion capture, human-computer interface, video matting, fire detection, etc.

The main application of background reconstruction is background subtraction, but other applications such as hole-filling in videos [Luo et al., 2016] have also been implemented. Efficient background reconstruction models are also necessary for unsupervised object detection and tracking [Jiang et al., 2020, Henderson and Lampert, 2020, Wu et al., 2021].

The model presented in this chapter starts from the classical assumption that the dynamic background of a scene can be modeled as a low dimensional manifold and uses an autoencoder to learn this manifold and perform dynamic background reconstruction. It then compares the input frame with the associated background predicted by the autoencoder to build the foreground segmentation mask. The main contributions of this chapter are the following :

- We implement a more robust loss function to train the autoencoder, which gives a high weight to reconstruction errors associated to background pixels and a low weight to reconstruction errors associated to foreground pixels, and shows better performance than the L_1 loss usually considered for this task.
- We train the autoencoder to provide a background reconstruction, but also a background noise estimation, which gives a pixelwise estimate of the uncertainty of the background prediction. This noise estimation map is used to adjust the threshold necessary to compute the background/foreground segmentation mask.
- We reduce the risk of overfitting by developping a method for detecting significant background changes and implementing an early stopping criterion using this method if the video shows a fixed background.

The chapter is structured as follows: We first review related work in section 2, then describe the proposed model in section 3. Experimental results are then provided in section 4.

4.4 Related work

Background subtraction methods can be split between supervised methods, which require labeled data, and unsupervised methods.

Supervised methods require labeled data as input, which are sets of pairs (X_k, M_k), where the image X_k is an image extracted from the sequence $X_1, .., X_N$ and the foreground mask M_k has to be provided by a human intervention. Supervised algorithms using linear methods such as as maximum margin criterion [Li et al., 2004, Diana and Bouwmans, 2010] or graph signal reconstruction methods [Giraldo and Bouwmans, 2020] have been proposed, but the current best performing supervised models use deep learning techniques with convolutional encoder-decoder structures [Lim and Yalim Keles, 2018, Lim and Keles, 2020, Mandal and Vipparthi, 2020], U-net structures [Rahmon et al., 2020, Mondéjar-Guerra et al., 2020] or GANs [Sultana et al., 2019, Zheng et al., 2020].

A spatio-temporal data augmentation strategy has been proposed [Tezcan et al., 2021] to improve generalization. One can also use as additional input to

the deep learning model the output of an unsupervised background subtraction model [Rahmon et al., 2020, Pardàs and Canet, 2021]. Although supervised models can reach very high accuracy results on a given video after labeling a significant number of frames of this video and training the model with these labeled data, their ability to generalize to new videos remain a major issue, and evaluations on unseen scenes lead to unfavorable results compared to unsupervised algorithms [Mandal and Vipparthi, 2020]. As a consequence, existing supervised models are not suited for real world applications where it is not possible to provide annotated data for each new input video.

One can classify **unsupervised methods** as statistical methods or reconstruction methods.

Statistical methods rely on a statistical modeling of the distribution of background pixel color values or other local features to predict whether a particular pixel is foreground or background. These statistical models can be parametric (univariate Gaussian [Wren et al., 1997], mixture of Gaussians [Stauffer and Grimson, 1999], clusters [Butler et al., 2005], Student's t-distributions [Mukherjee and Wu, 2012], Dirichlet process mixture models [Haines and Xiang, 2014], Poisson mixture models [Faro et al., 2011], asymmetric generalized Gaussian mixture models [Elguebaly and Bouguila, 2013], etc.) or non parametric (pixel value histograms [Zhang et al., 2009], kernel density estimation [Elgammal et al., 2000], codebooks [Kim et al., 2004], history of recently observed pixels [Barnich and Van Droogenbroeck, 2009, Hofmann et al., 2012], etc.). The efficiency of these methods can be increased by using as input not only the pixel color values, but also features attached to superpixels [Chen et al., 2019a] or local descriptors which are robust to illumination changes, such as SIFT [Pham and Smeulders, 2006], LBP or LBSP descriptors [St-Charles et al., 2015,St-Charles et al., 2016]. If the camera is static, the segmentation of moving objects on a scene can also be performed by evaluating the motion associated to each pixel, using optical flow or flux tensor models. The blobs produced by these models are generally very fuzzy, but can be used as input to more complex models [Bunyak et al., 2007, Wang et al., 2014a].

Reconstruction methods use a background reconstruction model to predict the color (or other features) of the background at a particular pixel. The difference between the current image and the predicted background is then computed and followed by a thresholding to decide whether a pixel is background or foreground. Pixelwise reconstruction models try to predict the value of a background pixel at a particular frame from the sequence of values of the pixel of the last frames using a filter, which can be a Wiener filter [Toyama et al., 1999], a Kalman filter [Ridder et al., 1995] or a Chebychev filter [Chang et al., 2004]. A global prediction of the background can also be performed using the assumption that the background frames form a low dimensional manifold, which motivates the use of dimensionality reduction techniques such as principal component analysis (PCA) [N.M. et al., 2000]. One can add to this approach a prior on the sparcity of the foreground objects by using a L_1 loss term applied to the foreground residuals, which leads to the development of models based on robust principal component analysis (RPCA) [Wright et al., 2009, Candès et al., 2011]. More complex norms and additional regularizers have been proposed to improve the performance of this approach [Mairal et al., 2010, Liu et al., 2015, Xin et al., 2015, Javed et al., 2017, Javed et al., 2019]. Non-linear dimensionality reduction using an autoencoder for background reconstruction

has been proposed in [Behnaz et al., 2021, Rezaei et al., 2020] and is further developed in the proposed model.

Several unsupervised models can be also combined to form a more accurate model, such as the IUTIS-5 models, which is an ensemble model combining 5 different unsupervised models [Bianco et al., 2017]. A background subtraction model can also be substantially improved by combining its results with the output of a supervised semantic segmentation model [Braham et al., 2018, Zeng et al., 2019].

It should be noted that the distinction between unsupervised models and supervised models is quite blurry considering that unsupervised models often use hyperparameters which are optimized using available annotated datasets, which can be considered as a form of supervision and may involve some overfitting. As a consequence, the evaluation of the robustness of an unsupervised model should take into account the number of hyperparameters involved, and be performed on a wide variety of videos.

Background noise estimation. Explicit background noise estimation for foreground segmentation has been introduced in [Hofmann et al., 2012]. Estimating the prediction uncertainty of a deep learning model is usually implemented using a negative log-likelihood loss function associated to a probabilistic model which includes a variance or concentration parameter [Nix and Weigend, 1994, Kendall and Gal, 2017, Bae et al., 2021, Moreau et al., 2022, Seitzer et al., 2022].

Several surveys [Bouwmans, 2014, Mondéjar-Guerra et al., 2020, Kalsotra and Arora, 2022, Mandal and Vipparthi, 2022, Zhao et al., 2022] discuss background reconstruction and background subtraction models.

4.5 Model description

The proposed model is a reconstruction model and has a general structure similar to the DeepPBM model [Behnaz et al., 2021]: We assume that the background frames form a low dimensional manifold and train an autoencoder to learn this manifold from the complete video. We however observe that the DeepPBM model described in [Behnaz et al., 2021] is not really unsupervised since it requires a significant engineering and optimization work for each new video, which is incompatible with any real-world application: The structure of the autoencoder and the number of latent variables have to be defined and fine-tuned on a scene by scene basis, which can be considered as a form of supervision. One also remarks that if the number of latent variables is too high, the autoencoder quiclky learns to reproduce the foreground objects, a phenomenon we call ovefitting, and fails to generate a proper background.

The model proposed in this chapter is fully unsupervised: It uses a constant set of hyperparameter, and the structure of the autoencoder, which depends on the size of the image and on the complexity of the background, is defined automatically without human supervision.

4.5.1 Reconstruction loss using background bootstrapping

We implement a reconstruction loss using background bootstrapping, adapted from the one described in Chapter 3. In the case of dynamic background re-

construction, this loss function allows to reduce the risk of overfitting to the foreground objects by giving a higher weight to background pixels than to foreground pixels during the optimization process. This loss is more robust to outliers than the L_1 loss which gives the same weight to small and large errors. The proposed reconstruction loss can be described by the following formulae: We note $x_{n,c,i,j}$ the pixel color value of the image X_n for the channel c at the position (i,j) with $1 \leq c \leq 3$,$1 \leq i \leq h$ and $1 \leq j \leq w$, and $\hat{x}_{n,c,i,j}$ the pixel value of the reconstructed background $\hat{X}_n$ for the same channel and position. The local L_1 error associated to the pixel (i,j) is

$$l_{n,i,j} = \sum_{c=1}^{3} |\hat{x}_{n,c,i,j} - x_{n,c,i,j}| \,. \tag{4.1}$$

The soft foreground masks and spatially smoothed soft foreground masks are defined by the equations

$$m_{n,i,j} = \tanh\left(\frac{l_{n,i,j}}{\tau_1}\right), \tag{4.2}$$

and

$$\tilde{m}_{n,i,j}(\hat{X}_n, X_n) = \frac{1}{(2k+1)^2} \sum_{l=-k,p=-k}^{l=k,p=k} m_{n,i+l,j+p} \,, \tag{4.3}$$

where τ_1 and r are positive hyperparameters and $k = \lfloor w/r \rfloor$. The associated pixel-wise weight $w_{n,i,j}^{\text{bootstrap}}$ is then defined as

$$w_{n,i,j}^{\text{bootstrap}} = e^{-\beta \tilde{m}_{n,i,j}}, \tag{4.4}$$

where β is another positive hyperparameter. The reconstruction loss of the auto-encoder is then computed by weighting the pixelwise L_1 losses $l_{n,i,j}$ using these bootstrap weights:

$$\mathcal{L}_{\text{rec}}(\hat{\mathcal{X}}, \mathcal{X}) = \frac{1}{Nhw} \sum_{n=1,i=1,j=1}^{N,h,w} w_{n,i,j}^{\text{bootstrap}} l_{n,i,j} \,. \tag{4.5}$$

The main differences between this loss function and the loss function defined in chapter 3 is that it is a one-to-one loss, whereas the loss defined in chapter 3 is one-to-many. It also does not use optical flow weights or abnormal image weights. Using optical flow weights would not allow to handle images taken from a moving camera, since it would give a low weight to all pixels associated to the moving background. We do not use abnormal image weights because we want the model to accurately reconstruct the background for each input image, which was not the case in chapter 3, which is dedicated to fixed background reconstruction.

4.5.2 Optimized thresholding using background noise estimation

We remark that the bootstrap pixel weights $w_{n,i,j}^{\text{bootstrap}}$ can be used to get an estimate of the level of background noise of a frame sequence, considering that

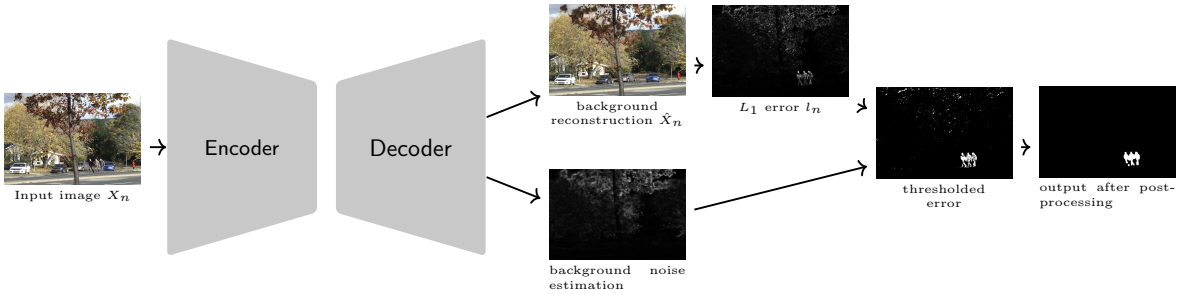


Figure 4.2: Schematic of the proposed model during inference (Error and noise images are normalized in the range [0,1].)

these weights are close to one when the associated pixel is a background pixel, and close to zero when this is not the case.

We therefore add a fourth output channel to the auto-encoder, which is dedicated to give an estimate $\hat{l}_{n,i,j}$ of the value of the L_1 error $l_{n,i,j}$ for each pixel (i, j) for the frame X_n (Fig. 4.2).

The associated loss function is weighted using the bootstrap weights in order to limit its scope to background regions:

$$\mathcal{L}_{\text{noise}} = \frac{1}{3Nhw} \sum_{n=1 i=1,j=1}^{N,h,w} w_{n,i,j}^{\text{bootstrap}} |\hat{l}_{n,i,j} - l_{n,i,j}| \,. \tag{4.6}$$

When the background is very noisy, the autoencoder is not able to predict accurately the value of a background pixel color. As a consequence, the expectation of $l_{n,i,j}$ is large, which leads to a high value of $\hat{l}_{n,i,j}$. One could consider that a more principled method would be to model the background noise as a Gaussian distribution and estimate the variance of this distribution by learning the weighted average L_2 error instead of the L_1 error, but we have empirically found that such an approach is not robust to the presence of foreground objects.

The autoencoder is trained using the sum of the reconstruction loss and the loss associated to the background noise estimation. The complete loss function is then

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{noise}}. \tag{4.7}$$

The gradients of the weights $w_{n,i,j}^{\text{bootstrap}}$ are not computed during the optimization process. We also do not use the gradient of $l_{n,i,j}$ in equation 4.6 because we do not want the quality of the background reconstruction be impacted by the background noise estimation optimization process.

In order to set the pixelwise threshold $\tau_{n,i,j}$ associated to the pixel (i, j) of the frame X_n and necessary to compute the background/foreground segmentation mask, we also take into account the average illumination $\hat{I}_n$ of the reconstructed background $\hat{X}_n$, as defined by the formula

$$\hat{I}_n = \frac{1}{3hw} \sum_{c=1,i=1,j=1}^{3,h,w} |\hat{x}_{n,c,i,j}|. \tag{4.8}$$

The threshold $\tau_{n,i,j}$ is then set according to the formula

$$\tau_{n,i,j} = \alpha_1 \hat{I}_n + \alpha_2 \hat{l}_{n,i,j}, \tag{4.9}$$

where α_1 and α_2 are two positive hyperparameters.The α_1 hyperparameter can then be interpreted as the threshold applicable to a scene showing a noiseless white background. The motivation of the second term is that if the background noise is high at some pixel, we have to increase the associated threshold for background/foreground segmentation in order to prevent the misclassification of background pixels as foreground caused by background noise.

For a given frame sequence $X_1, ..., X_n$ and a reconstructed background sequence $\hat{X}_1, ..., \hat{X}_n$, we then compute the foreground mask M_n before post-processing using the thresholding rule $M_{n,i,j} = 1$ if and only if $l_{n,i,j} > \tau_{n,i,j}$.

A post-processing is then applied in order to remove rain drops, snow flakes, and other spurious detections. It is composed of two morphological operations: a morphological closing using a 5×5 square structural element, followed by a morphological opening with a 7×7 square structural element.

4.5.3 Detecting significant background changes

The improved reconstruction loss function introduced in 4.5.1 reduces the risk of overfitting, but is not able to prevent it completely. We observe that the risk of overfitting increases when the number of optimization iterations and the number of parameters of the network increase. This is a significant issue because sequences showing background changes require a high number of training iterations and a model with a large number of parameters. In order to prevent overfitting, the number of training iterations and the complexity of the model are therefore adjusted to the complexity of the backgrounds sequence.

The main challenge here is to estimate without any human supervision whether the video shows substantial background changes or not. Such a task, which is very easy for a human, is far from trivial for a computer. For example, simply taking the variance of the various frames does not allow to estimate the complexity of the background changes because this variance will generally be dominated by foreground objects appearing in the video. More generally, it appears that in order to estimate the importance of the background changes, it is necessary to remove the foreground objects from the estimation process. We observe however that the proposed model can be used to perform this task. We then first train the model for a fixed small number N_{eval} of iterations, which is however sufficient to get a rough evaluation of the background changes. Using this trained model, we compute B_{eval} reconstructed backgrounds $\hat{X}_n$ using frames X_n sampled randomly from the sequence $\mathcal{X}$. Although these backgrounds estimates $\hat{X}_n$ are not accurate, we are confident that they do not show any foreground objects since a low number of iterations have been performed, so that the risk of overfitting is very low. We then compute the temporal median $\hat{X}$ of these backgrounds and compare this median background with the reconstructed backgrounds $\hat{X}_n$, computing soft masks $m_{n,i,j}$ following the same process as in formula 4.1 and 4.2. We then consider the average soft mask value over the B_{eval} reconstructed backgrounds

$$\bar{m} = \frac{1}{B_{\text{eval}} hw} \sum_{n,i,j}^{B_{\text{eval}},h,w} m_{n,i,j}. \tag{4.10}$$

If $\bar{m}$ is higher than a threshold τ_0, we consider that the background is a complex background. The partially trained model is discarded, a new autoencoder is

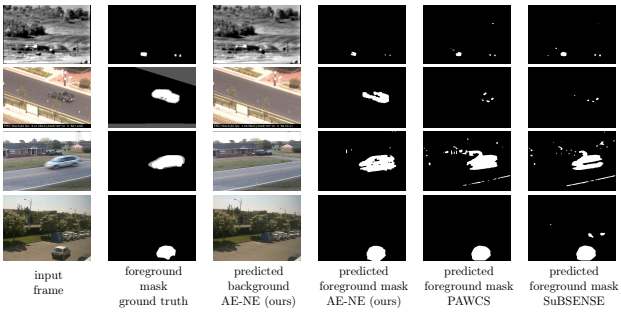


Figure 4.3: Examples of background reconstruction and foreground segmentation produced using the proposed model and comparison with PAWCS and SuBSENSE.

created with more parameters and the number of training iterations is set to N_{complex} with a minimum of E_{complex} epochs for very long sequences. If this ratio is lower than τ_0, we consider that the background is a simple background, keep the partially trained model, and finish the training, with a total number of training iterations set to N_{simple}. The autoencoder structures for simple and complex backgrounds are described in the supplementary material.

4.6 Experimental results

4.6.1 Evaluation method

We consider the CDnet 2014, LASIESTA and BMC 2012 benchmark datasets for background subtraction. We use the public implementations of the algorithms PAWCS [St-Charles et al., 2016] and SuBSENSE [St-Charles et al., 2015] provided with the BGS library [Sobral, 2013] to get baseline performance estimates for these methods when they are not available. We rely on published results for the other state of the art methods which do not provide public implementations.

We use the F-measure as main evaluation criteria. To compute the F-measure associated to a sequence of foreground masks predictions $M_1, .., M_n$, we first compute the sums TP, TN, FP, FN of the true positives, true negatives, false positives and false negatives associated to the sequence of masks $M_1, .., M_n$, and then compute the F-measure associated to this sequence as the harmonic mean of precision and recall.

We provide in Figure 4.3 some samples of background reconstruction, with the associated predicted foreground mask, and a comparison with foreground masks obtained using PAWCS and SuBSENSE. Other samples are provided in the supplementary material.

4.6.2 CDnet 2014 dataset

The CDnet 2014 dataset [Wang et al., 2014b] is composed of 53 videos, for a total of 153 278 frames, selected to cover the various challenges which have to be addressed for background subtraction: dynamic background (scenes with water or trees), camera jitter, intermittent object motion, presence of shadows, images captured by infrared cameras, challenging weather (snow, fog), images

Table 4.1: Comparison of top BGS algorithms according to the per-category F-measures on CDnet-2014

Method	Bad weather	Base-line	Camera jitter	Dynamic backgr.	Int. obj. motion	Low framerate	Night	PTZ	Shadow	Thermal	Turbu-lence	Overall
models using frame annotations (full supervision):												
FgSegNet v2 [Lim and Keles, 2020]	0.9904	0.9978	0.9971	0.9951	0.9961	0.9336	0.9739	0.9862	0.9955	0.9938	0.9727	0.9847
BSUV-Net 2.0 [Tezcan et al., 2021]	0.8844	0.9620	0.9004	0.9057	0.8263	0.7902	0.5857	0.7037	0.9562	0.8932	0.8174	0.8387
model using pretrained semantic segmentation model:												
SemanticBGS [Braham et al., 2018]	0.8260	0.9604	0.8388	0.9489	0.7878	0.7888	0.5014	0.5673	0.9478	0.8219	0.6921	0.7892
models using no frame annotation or pretrained model:												
AE-NE (ours)	0.8337	0.8959	0.9230	0.6225	0.8231	0.6771	0.5172	0.8000	0.8947	0.7999	0.8382	0.7841
IUTIS-5 [Bianco et al., 2017]	0.8248	0.9567	0.8332	0.8902	0.7296	0.7743	0.5290	0.4282	0.9084	0.8303	0.7836	0.7717
WisenetMD [Lee et al., 2019]	0.8616	0.9487	0.8228	0.8376	0.7264	0.6404	0.5701	0.3367	0.8984	0.8152	0.8304	0.7535
SuBSENSE [St-Charles et al., 2015]	0.8619	0.9503	0.8152	0.8177	0.6569	0.6445	0.5599	0.3476	0.8986	0.8171	0.7792	0.7408
PAWCS [St-Charles et al., 2016]	0.8152	0.9397	0.8137	0.8938	0.7764	0.6588	0.4152	0.4615	0.8913	0.8324	0.6450	0.7403
C-EFIC [Allebosch et al., 2016]	0.7867	0.9309	0.8248	0.5627	0.6229	0.6806	0.6677	0.6207	0.8778	0.8349	0.6275	0.7307
MSCL [Javed et al., 2017]	0.83	0.87	0.83	0.85	0.80	n/a	n/a	n/a	0.82	0.80	0.80	n/a
B-SSSR [Javed et al., 2019]	0.92	0.97	0.93	0.95	0.74	n/a	n/a	n/a	0.93	0.86	0.87	n/a

captured with a low frame rate, night images, images filmed by a pan-tilt-zoom camera, air turbulence. Ground truth foreground segmentation masks are provided for all frames of the dataset, with specific labels for shadow pixels which are not considered in the F-measure computation. We provide in Table 4.1 the F-measure results per category of the proposed model for each category of the CDnet 2014 dataset, with a comparison with the results obtained by other unsupervised models.

The proposed model gets a higher average F-measure on the CDnet 2014 dataset than all published unsupervised models, including ensemble models such as IUTIS-5, with an average F-measure of 0.784. One can observe a significant improvement in accuracy with the proposed model in the "pan-tilt-zoom" (PTZ) category with an average F-measure of 0.800 on this category. To our best knowledge, the proposed model is the first able to correctly handle videos taken from a moving camera.

4.6.3 LASIESTA dataset

The LASIESTA dataset [Cuevas et al., 2016] is composed of 48 videos grouped in 14 categories, for a total of 18 425 video frames. All frames are provided with ground truth pixel labels, with a specific label for pixels associated to stopped moving objects which are excluded from the F-measure computation. These videos are very short (The average number of frames per video is 383), which is challenging for the proposed deep-learning based model. We provide in Table 4.2 the average F-measure results of the proposed model for all 14 categories. Out of the 48 videos of the dataset, 4 videos are taken with a moving camera (categories IMC and OMC), and 24 videos include simulated camera motion (categories ISM and OSM). These 28 videos which include real or simulated camera motion are very difficult for existing background subtraction models and to our best knowledge, no paper has ever published category-wise evaluation results for these videos. In order to allow a comparison with these published results, we therefore also provide the average F-measure over the 10 categories showing only videos taken from a fixed camera. We observe that the proposed model performs better than available unsupervised algorithms on static scenes, and with a significant improvement on scenes where the camera is moving.

Table 4.2: Average per category of video F-measures on LASIESTA (sources : [Cuevas et al., 2016], [Berjón et al., 2018], authors experiments for PAWCS and SuBSENSE)

Method	static camera										moving camera or simulated motion				Average. 10 categ.	Average. 14 categ.
	ISI	ICA	IOC	IIL	IMB	IBS	OCL	ORA	OSN	OSU	IMC	ISM	OMC	OSM		
AE-NE (ours)	0.91	0.88	0.91	0.81	0.92	0.79	0.94	0.80	0.82	0.91	0.83	0.79	0.86	0.89	**0.87**	**0.86**
PAWCS [St-Charles et al., 2016]	0.90	0.88	0.90	0.79	0.81	0.79	0.96	0.93	0.69	0.82	0.48	0.77	0.43	0.75	0.85	0.78
SuBSENSE [St-Charles et al., 2015]	0.90	0.89	0.95	0.65	0.77	0.73	0.92	0.90	0.81	0.79	0.33	0.70	0.31	0.65	0.83	0.73
Cuevas [Berjón et al., 2018]	0.88	0.84	0.78	0.65	0.93	0.66	0.93	0.87	0.78	0.72	n/a	n/a	n/a	n/a	0.81	n/a
Haines [Haines and Xiang, 2014]	0.89	0.89	0.92	0.85	0.84	0.68	0.83	0.89	0.17	0.86	n/a	n/a	n/a	n/a	0.78	n/a
Maddalena [Maddalena and Petrosino, 2012]	0.95	0.86	0.95	0.21	0.91	0.40	0.97	0.90	0.81	0.88	n/a	n/a	n/a	n/a	0.78	n/a
Maddalena [Maddalena and Petrosino, 2008b]	0.87	0.85	0.91	0.61	0.76	0.42	0.88	0.84	0.58	0.80	n/a	n/a	n/a	n/a	0.75	n/a

Table 4.3: Comparison of top unsupervised BGS algorithms according to the video F-measure on BMC 2012

Method	Video 001	Video 002	Video 003	Video 004	Video 005	Video 006	Video 007	Video 008	Video 009	Average 9 videos
F-measure (standard definition)										
AE-NE (ours)	0.81	0.72	0.78	0.78	0.60	0.73	0.32	0.84	0.77	**0.71**
PAWCS [St-Charles et al., 2016]	0.70	0.58	0.85	0.72	0.27	0.79	0.58	0.74	0.80	0.67
SuBSENSE [St-Charles et al., 2015]	0.70	0.62	0.83	0.69	0.21	0.76	0.53	0.68	0.83	0.65
F-measure (using BMC evaluation tool)										
AE-NE (ours)	0.90	0.86	0.89	0.89	0.80	0.87	0.51	0.92	0.89	0.84
PAWCS [St-Charles et al., 2016]	0.86	0.77	0.93	0.86	0.66	0.89	0.79	0.87	0.90	0.84
SubSENSE [St-Charles et al., 2015]	0.85	0.80	0.92	0.85	0.68	0.87	0.75	0.84	0.91	0.83
DeepPBM [Behnaz et al., 2021]	0.73	0.86	0.94	0.90	0.71	0.81	0.70	0.76	0.69	0.78
G-LBM [Rezaei et al., 2020]	0.73	0.85	0.93	0.91	0.71	0.85	0.70	0.76	0.63	0.79
MSCL-FL [Javed et al., 2017]	0.84	0.84	0.88	0.90	0.83	0.80	0.78	0.85	0.94	0.86
B-SSSR [Javed et al., 2019]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	**0.88**

4.6.4 BMC 2012 dataset

The BMC dataset [Vacavant et al., 2013] contains 9 videos showing real scenes taken from static cameras and including the following challenges: shadows, snow, rain, presence of trees or big objects. Three of these sequences are very long (32 965, 117 149 and 107 815 frames). For fair comparison with other published results for this dataset, we provide the F-measure results for our model obtained using the usual F-measure definition described in 4.6.1, but also the results obtained using the executable evaluation tool provided with the dataset which does not use the same definition of the F-measure [Vacavant et al., 2013]. We compute SuBSENSE and PAWCS results on this dataset and provide published evaluation results for other models in Table 4.3.

We observe that the proposed model gets again a better average F-measure than PAWCS and SuBSENSE on this dataset using the standard definition of the F-measure.

4.6.5 Non-video image datasets : Clevrtex, ObjectsRoom, ShapeStacks

The proposed model, which does not use any temporal information, can be adapted to perform background reconstruction and foreground segmentation on some image datasets which are not extracted from video sequences. We have tested this approach on three synthetic image datasets: Clevrtex [Karazija et al., 2021], ShapeStacks, [Groth et al., 2018] and ObjectsRoom [Kabra et al., 2019]. We use on ShapeStacks and ObjectsRoom the same preprocessing as in

Table 4.4: F-Measure on the Clevrtex, ShapeStacks and ObjectsRoom datasets

dataset	image size	number of frames training set	number of frames test set	average F-measure on test set
Clevrtex	128×128	40000	5000	0.78
ObjectsRoom	64×64	980000	20000	0.84
ShapeStacks	64×64	217888	46656	0.83

[Engelcke et al., 2021]. Although each image of these datasets shows a different background, the model is able to recognize that all the backgrounds appearing in a given dataset lie in a low dimensional manifold, which is the case because they have been generated using the same method. These datasets are provided with segmentation annotations for each object appearing in the scenes, which we converted to binary foreground segmentation masks in order to compute the F-measure of the predicted foreground masks.

Considering that on these datasets the risk of overfitting is very low and the background complexity is very high, we substantially increased the number of iterations, which is set to 500 000. We do not use morphological post-processing on the ShapeStacks and ObjectsRoom datasets, because these images have a very low resolution (64×64). We provide in Table 4.4 the average F-measure obtained on the test sets of these datasets after training on the associated training sets, and in Figure 4.4 some image samples. To our best knowledge, no other model is able to perform background reconstruction on these datasets.

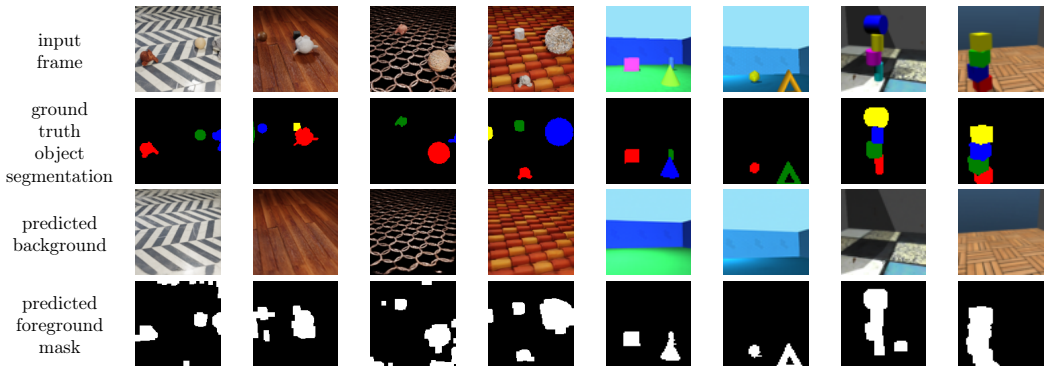


Figure 4.4: Examples of background reconstruction and foreground segmentations on the datasets Clevrtex (columns 1-4), ObjectsRoom (columns 5-6) and ShapeStacks (columns 7-8)

4.6.6 Robustness to domain shift and fine-tuning

The proposed model is a batch model. In order to see whether it could be adapted for real-time applications, we studied whether a trained model could perform background reconstruction on new unseen images of the scene which do not belong exactly to the same distribution as the images used for training due to various possible domain shifts such as unseen illumination changes. We then have performed the following experiment: We have split each of the 53 videos provided in the CDnet dataset in two videos of equal lengths. The first half of

Table 4.5: F-measure results obtained on the CDnet dataset with a model pretrained using the first half of each video as training set, and fine-tuned on the last half using various numbers of fine-tuning iterations. Test results are for the last half of each video.

	Bad weather	Base-line	Camera jitter	Dynamic backgr.	Int. obj. motion	Low framerate	Night	PTZ	Shadow	Thermal	Turbu-lence	Overall	no pretraining
no fine-tuning	0.8114	0.8660	0.8768	0.3845	0.4199	0.5732	0.3998	0.2426	0.7371	0.5872	0.6447	0.5948	
100 iterations	0.8137	0.9063	0.9520	0.5846	0.5956	0.5891	0.4789	0.4723	0.9276	0.7639	0.6639	0.7044	0.4918
200 iterations	0.8078	0.9105	0.9543	0.6111	0.6536	0.5859	0.4977	0.4969	0.9316	0.7849	0.7523	0.7261	0.5658
400 iterations	0.8080	0.9125	0.9560	0.6309	0.7298	0.5842	0.5137	0.5465	0.9326	0.7880	0.8560	0.7507	0.6218
800 iterations	0.8104	0.8965	0.9577	0.6348	0.8212	0.5946	0.5420	0.6403	0.9293	0.7828	0.8763	0.7714	0.6934

each video is used to train the autoencoder, and the second half is used as a test dataset. The results of this experiment are provided in Table 4.5 and show stable results on three categories (baseline, bad weather, camera jitter) which do not show noticeable domain shifts, but a significant worsening on the other categories.

We then adopt the pretrain/fine-tune paradigm, consider the models trained on the first half of the videos as pretrained models, and study how many fine-tuning iterations using images randomly sampled from the second half of the videos are necessary to get competitive test results. We observe that the number of required iterations is very low compared to the number of iterations necessary for a full training, and conclude that a trained model is not robust to domain shifts, but can be quickly updated with a small number of fine-tuning iterations.

4.6.7 Implementation details

The proposed model is implemented using Python and the Pytorch framework. The associated code is available on the Github platform. Optimization is performed using the Adam optimizer with a learning rate of 5.10^{-4} and batch size equal to 32. The learning rate is divided by 10 when the number of optimization or fine-tuning iterations reaches 80% of the total number of iterations. The most important hyperparameters β, r and τ_1, which are associated to the loss function, are set to the values recommended in chapter 3 i.e. $\beta = 6$, $r = 75$, $\tau_1 = 0.25$. The other hyperparameter values, which are related to the segmentation threshold and the detection and management of complex background changes, were found empirically using manual hyperparameter tuning. We then set $\alpha_1 = 96/255$, $\alpha_2 = 7$, $N_{\text{eval}} = 2000$, $B_{\text{eval}} = 480$, $\tau_0 = 0.24$, $N_{\text{simple}} = 2500$, $N_{\text{complex}} = 24000$, $E_{\text{complex}} = 20$.

For non-video dataset experiments, which take small images (64×64 and 128×128) as inputs, the batch size and learning rate are increased to 128 and 2.10^{-3} and the number of iterations N_{complex} is set to 500 000. The other hyperparameters remain the same. The autoencoder architecture is described in the supplementary material.

4.6.8 Computation time

We provide in Table 4.6 some computation time measurements, obtained using an AMD EPYC 7402 2,8 GHz CPU and a Nvidia RTX 3090 GPU. The inference and training times of the proposed model depend on the size of the image and the complexity of the background. The inference speed is between 50 frames per

Table 4.6: Computation time of the proposed model, PAWCS and SubSENSE for some sequences of the CDnet and BMC datasets

sequence name	highway	Video 009	blizzard	zoomin zooomout	continuous pan
image size	240x320	288x352	480x720	240x320	480x704
number of frames	1700	107817	7000	1130	1700
background complexity	simple	simple	simple	complex	complex
computation times (seconds)					
AE-NE (proposed model)					
- training	92	114	394	1443	7175
- backgrounds and masks generation	7	560	139	5	33
SuBSENSE	92	7161	1586	65	471
PAWCS	158	11290	2311	164	980

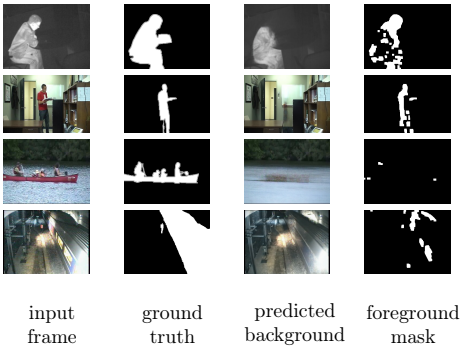


Figure 4.5: Failure cases due to overfitting on the datasets CDnet 2014 and BMC 2012: sequences "library", "office", "canoe" and "video007"

second and 240 frames per second. The time necessary to perform 100 training iterations is between 3,5 and 27 seconds.

4.6.9 Limitations

This model is not suited for night videos, considering the low score obtained on this category on the CDnet dataset. One also notes that although the model is able to handle correctly small objects staying still for a long time, as shown by the good results obtained the intermittent object category of the CDnet dataset, it suffers from overfitting when large foreground objects stay still (or appear to stay still) for a long time in a frame sequence. Out of the 110 tested videos contained in the datasets CDnet, LASIESTA and BMC, we observed this problem on 4 videos: "office", "library" and "canoe" in the CDnet dataset, and "video007" in the BMC dataset (Fig. 4.5). The proposed model should then not be used when the video is expected to show large objects staying still for a long time. This model is a batch model and adapting it to real-time applications requires further work in order to reduce the latency caused by the fine-tuning iterations described in section 4.6.6.

4.6.10 Ablation study

In order to assess the impact of the various model features described in this paper, we have implemented several modifications of the proposed model and mea-

Table 4.7: Evaluation of various ablations of the proposed model

model description	average F-measure on the CDnet dataset	evolution vs reference model
proposed model (reference)	0.7841	
modified models :		
- no bootstrap weights ($w_{n,i,j}^{\text{bootstrap}}$ set to 1)	0.2771	-64,6 %
- inference without using the background noise estimation (α_2 set to 0)	0.6220	-20.7 %
- $w_{n,i,j}^{\text{bootstrap}}$ set to 1 and α_2 set to 0	0.4557	-41,9%
- training with L_2 reconstruction loss, α_2 set to 0	0,3384	-56,8 %
- inference without morphological post-processing	0.7170	-8.5%
- all backgrounds are considered as simple (τ_0 set to 1)	0,7397	-5,6 %
- using optical flow weights as in chapter 3	0,7701	-1,8%
- using abnormal image weights as in chapter 3	0,7690	-1,9%

sured the average F-measure (FM) of these models on the CDnet2014 dataset. The results of these experiments are provided in Table 4.7. They show that the design of the loss function and the use of the background noise estimation layer have a substantial positive impact on the accuracy of the model. The improvement associated to post-processing is also significant, as already observed for other unsupervised background subtraction methods [Shahbaz et al., 2015]. The model remains competitive on CDnet if the background complexity of all frames sequence is set to simple, an option which may be considered if training computation time is an issue.

4.7 Conclusion of chapter 4

We have proposed in this chapter a new fully unsupervised dynamic background reconstruction and foreground segmentation model which does not use any temporal or motion information and is on average more accurate than available unsupervised models for background subtraction. The main strength of the proposed model is that it is able to perform background reconstruction on videos taken from a moving camera.

If the various objects appearing in a scene do not touch each other, a background/foreground segmentation model can be used to get a segmentation of these objects using connected component labeling algorithms such as the classical Rosenfeld and Pfalz labeling algorithm [Rosenfeld and Pfaltz, 1966]. However, if a scene shows objects which touch or occlude each other, which is often the case in traffic videos, a connected component labeling algorithm will fail to distinguish these objects, and the development of a model dedicated to multi-object detection and segmentation appears necessary. This topic is the subject of chapter 5.

4.8 Appendix to chapter 4

4.8.1 Autoencoder architecture

The autoencoder is deterministic and takes as input a RGB image of size $h \times w$, and produces a RGB image (3 channels) and an error estimation map of the same size (1 channel).

The encoder and decoder structures in the proposed model are computed dynamically using as input the size (height h and width w) of the input frames of the dataset. The number of latent variables produced by the encoder is fixed to 16.

We use a fully convolutional autoencoder architecture, which appears to be more robust to overfitting than architectures including fully connected layers or locally connected layers. We add two fixed positional encoding channels as inputs to all layers of the encoder and the decoder, one channel coding for the horizontal coordinates, the other one for the vertical coordinates .

The encoder is a sequence of blocks composed of a convolution layer with kernel size 5, stride 3 and padding equal to 2, followed by a group normalization layer and a CELU nonlinearity layer. The generator is a symmetric sequence of blocks composed of transpose convolution layers with kernel size 5 and stride 3 and padding equal to 2 followed by group normalization and a CELU nonlinearity, except for the last layer where the transpose convolution layer is followed by a sigmoid to generate the final image. The number of layers of the encoder and the decoder is then equal to 5 or 6 depending on the image size (assuming that the maximum of the image height and image width is in the range $200 - 1000$). The number of channels per convolutional layer is fixed according to Table 4.8, depending on the image size and the background complexity.

Table 4.8: Number of channels for each layer of the encoder and decoder (excluding positional encoding input channels)

background complexity	image size max(h,w)	Encoder	Decoder
simple	200-405	(3,64,160,160,32,16)	(16,32,256,256,144,4)
simple	406-1000	(3,64,160,160,160,32,16)	(16,32,256,512,256,144,4)
complex	200-405	(3,64,160,160,16,16)	(16,16,640,640,144,4)
complex	406-1000	(3,64,160,160,160,16,16)	(16,16,640,1280,640,144,4)

These channel distributions are motivated by the fact that a larger number of parameters is required in the generator in order to handle complex backgrounds, but that we have experimentally observed that a large number of channels in the last layer of the encoder and the first layer of the decoder increases the risk of overfitting on foreground objects, so that reducing this number for long training schedule is necessary to improve the robustness of the auto-encoder with respect to the risk of overfitting. For example, we have measured that increasing the numbers of channels in the last hidden layer of the encoder and first hidden layer of the decoder to 160 and 256 leads to de 2,3 % degradation of the average F-Measure on the CDnet dataset.

For non-video dataset experiments, which handle small images, we use a smaller stride, set to 2 instead of 3. The autoencoder architectures for 64×64 images (ShapeStacks and ObjectRooms datasets) and 128×128 images (Clevrtex dataset) are described in Table 4.9 and 4.10.

4.8.2 Additional implementation details

The datasets and preprocessing codes for CLEVRTEX, Shapestacks and ObjectsRoom were downloaded from the following public repositories:

Table 4.9: autoencoder architecture for 64×64 images

Encoder

Layer	Size	Ch	Stride	Norm./Act.
Input	64	3		
Conv 5×5	32	64	2	GroupNorm/CELU
Conv 5×5	16	160	2	GroupNorm/CELU
Conv 5×5	8	320	2	GroupNorm/CELU
Conv 5×5	4	160	2	GroupNorm/CELU
Conv 4×4	2	16	2	GroupNorm/CELU
Conv 2×2	1	16	1	GroupNorm/CELU

Decoder

Layer	Size	Ch	Stride	Norm./Act.
Input	1	16		
Conv Transp 2×2	2	16	1	GroupNorm/CELU
Conv Transp 4×4	4	640	2	GroupNorm/CELU
Conv Transp 5×5	8	1280	2	GroupNorm/CELU
Conv Transp 5×5	16	640	2	GroupNorm/CELU
Conv Transp 5×5	32	144	2	GroupNorm/CELU
Conv Transp 5×5	64	4	2	
Sigmoid	64	4		

Table 4.10: autoencoder architecture for 128×128 images

Encoder

Layer	Size	Ch	Stride	Norm./Act.
Input	128	3		
Conv 5×5	64	64	2	GroupNorm/CELU
Conv 5×5	32	320	2	GroupNorm/CELU
Conv 5×5	16	640	2	GroupNorm/CELU
Conv 5×5	8	640	2	GroupNorm/CELU
Conv 5×5	4	320	2	GroupNorm/CELU
Conv 4×4	2	16	2	GroupNorm/CELU
Conv 2×2	1	16	1	GroupNorm/CELU

Decoder

Layer	Size	Ch	Stride	Norm./Act.
Input	1	16		
Conv Transp 2×2	2	16	1	GroupNorm/CELU
Conv Transp 4×4	4	320	2	GroupNorm/CELU
Conv Transp 5×5	8	640	2	GroupNorm/CELU
Conv Transp 5×5	16	1280	2	GroupNorm/CELU
Conv Transp 5×5	32	640	2	GroupNorm/CELU
Conv Transp 5×5	64	144	2	GroupNorm/CELU
Conv Transp 5×5	128	4	2	
Sigmoid	128	4		

- https://www.robots.ox.ac.uk/~vgg/data/clevrtex/
- https://ogroth.github.io/shapestacks/
- https://github.com/deepmind/multi_object_datasets

4.8.3 Additional image samples

We provide in Figs. 4.6 to 4.12 additional samples of background reconstruction and foreground segmentation obtained using the proposed model.

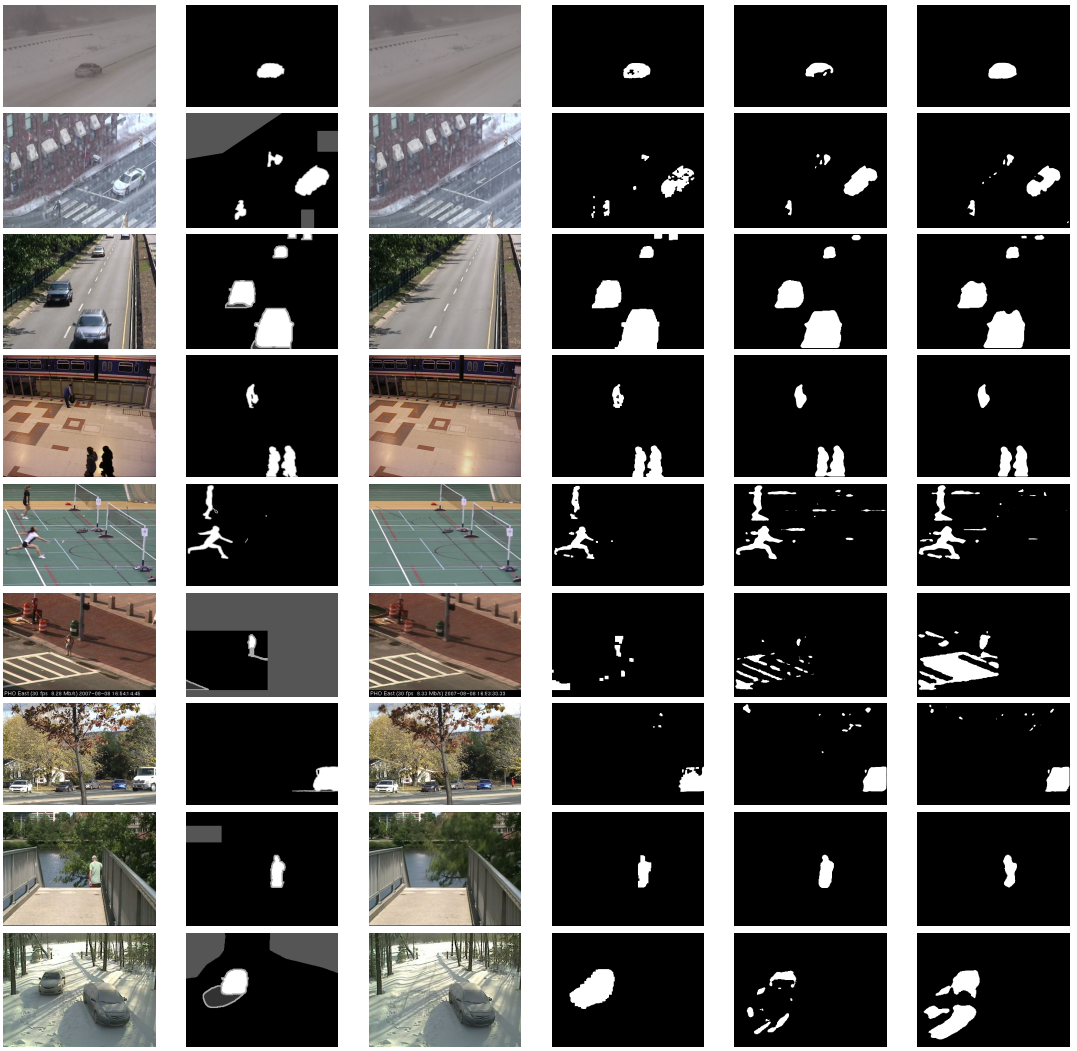


Figure 4.6: Examples of background reconstruction and foreground segmentation on the CDnet 2014 dataset produced using the proposed model and comparison with PAWCS and SuBSENSE.

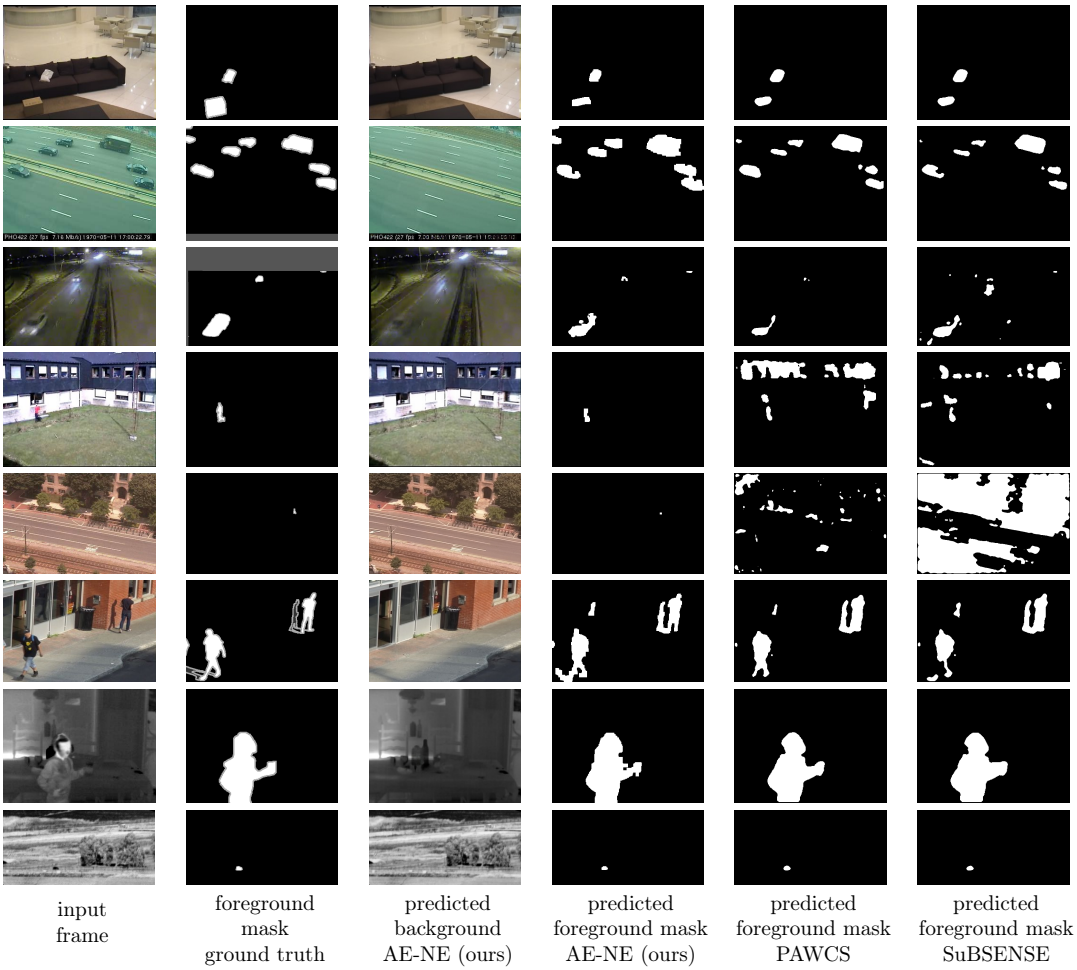


Figure 4.7: Examples of background reconstruction and foreground segmentation on the CDnet 2014 dataset produced using the proposed model and comparison with PAWCS and SuBSENSE.

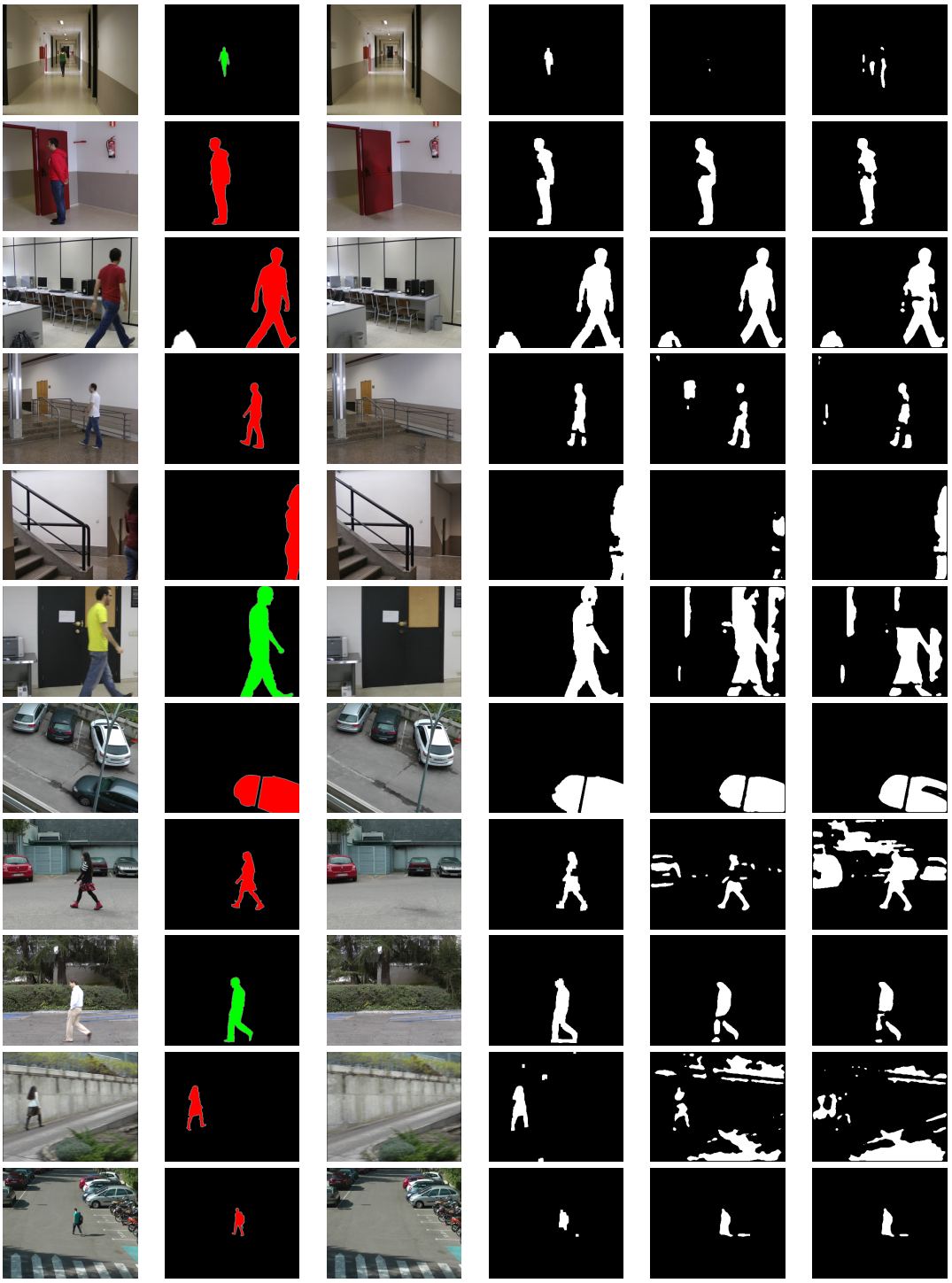


Figure 4.8: Examples of background reconstruction and foreground segmentation on the LASIESTA dataset produced using the proposed model and comparison with PAWCS and SuBSENSE.

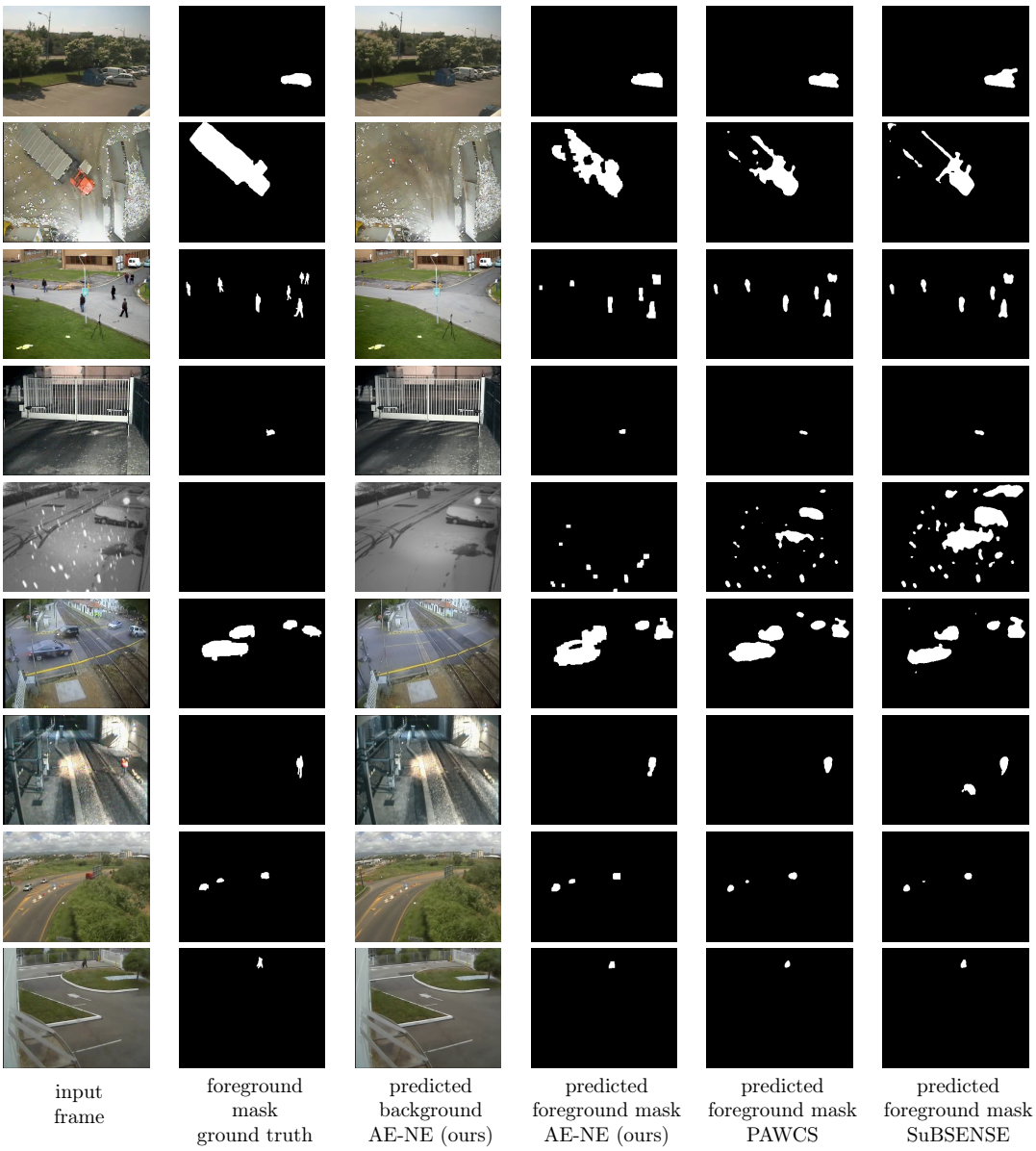


Figure 4.9: Examples of background reconstruction and foreground segmentation on the BMC 2012 dataset produced using the proposed model and comparison with PAWCS and SuBSENSE.

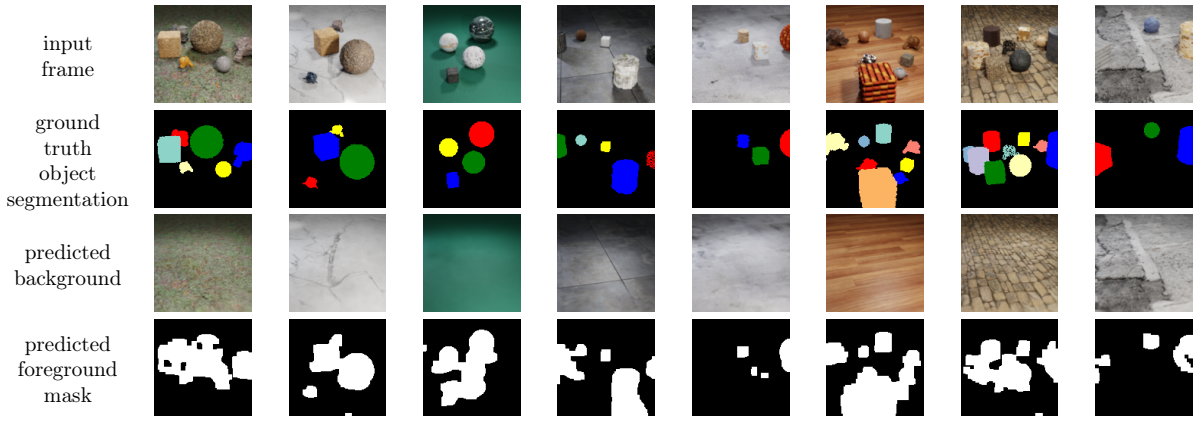


Figure 4.10: Examples of background reconstruction and foreground segmentation on Clevrtex dataset.

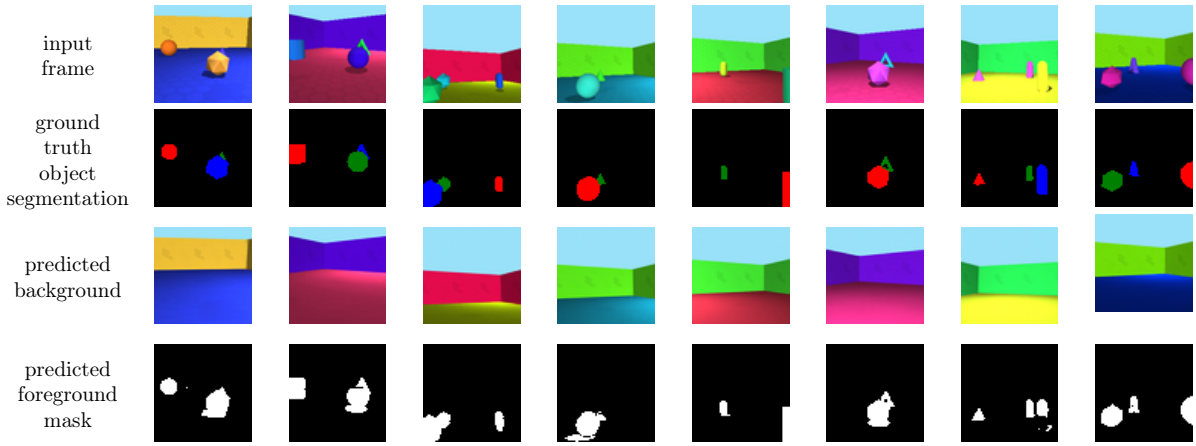


Figure 4.11: Examples of background reconstruction and foreground segmentation on ObjectsRoom dataset.

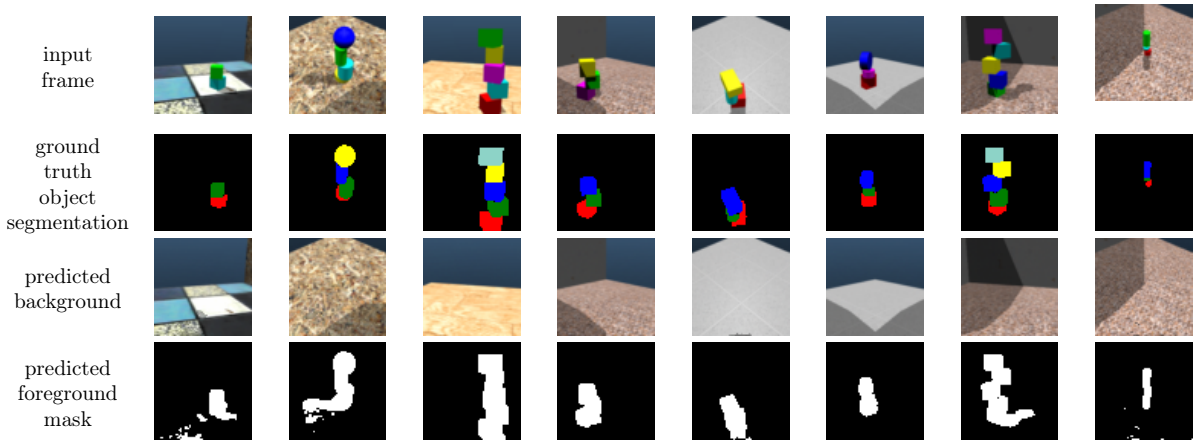


Figure 4.12: Examples of background reconstruction and foreground segmentation on ShapeStacks dataset.

Chapter 5

Unsupervised object-centric representation learning and multi-object segmentation

5.1 Résumé en français

Nous présentons dans ce chapitre une nouvelle architecture pour l'apprentissage de représentations centrées sur les objets et la détection et la segmentation multi-objets, qui utilise un mécanisme d'attention équivariant aux translations pour prédire les coordonnées des objets présents dans la scène et pour associer un vecteur de caractéristiques à chaque objet. Un encodeur de type transformer gère les occultations et les détections redondantes, et un auto-encodeur convolutionnel est en charge de la reconstruction de l'arrière-plan. Nous montrons que cette architecture dépasse significativement l'état de l'art sur les benchmarks synthétiques complexes et donnons quelques exemples d'application sur des vidéos non synthétiques issues de caméras de circulation.

5.2 Abstract

We introduce a new architecture for unsupervised object-centric representation learning and multi-object detection and segmentation, which uses a translation-equivariant attention mechanism to predict the coordinates of the objects present in the scene and to associate a feature vector to each object. A transformer encoder handles occlusions and redundant detections, and a convolutional autoencoder is in charge of background reconstruction. We show that this architecture significantly outperforms the state of the art on complex synthetic benchmarks and provide examples of applications to real-world traffic videos.

5.3 Introduction

We consider in this chapter the tasks of object-centric representation learning and unsupervised object detection and segmentation: Starting from a dataset

of images showing various scenes cluttered with objects, our goal is to build a structured object-centric representation of these scenes, i.e. to map each object present in a scene to a vector representing this object and allowing to recover its appearance and segmentation mask. This task is very challenging because the objects appearing in the images may have different shapes, locations, colors or textures, can occlude each other, and we do not assume that the images share the same background. However the rewards of object-centric representations could be significant since they allow to perform complex reasoning on images or videos [Ding et al., 2021, Tang et al., 2022] and to learn better policies on downstream tasks involving object manipulation or localization [Veerapaneni et al., 2020, Zadaianchuk et al., 2021]. The main issue with object-representation learning today is however that existing models are able to process synthetic toy scenes with simple textures and backgrounds but fail to handle more complex or real-world scenes [Karazija et al., 2021].

We propose to improve upon this situation by introducing a translation-equivariant and attention-based approach for unsupervised object detection, so that a translation of the input image leads to a similar translation of the coordinates of the detected objects, thanks to an attention map which is used not only to associate a feature vector to each object present in the scene, but also to predict the coordinates of these objects.

The main contributions of this chapter are the following:

- We propose a theoretical justification for the use of attention maps and soft-argmax for object localization.
- We introduce a new translation-equivariant and attention-based object detection and segmentation architecture which does not rely on any spatial prior.
- We show that the proposed model substantially improves upon the state of the art on unsupervised object segmentation on complex synthetic benchmarks.

The chapter is organized as follows: In section 5.4, we provide some theoretical motivation for using attention maps and soft-argmax for object localization. In section 5.5, we review related work on unsupervised object instance segmentation. In section 5.6 we describe the proposed model. Experimental results are then provided in section 5.7.

5.4 Motivation for using attention maps and soft-argmax for object localization

It is widely recognized that the success of convolutional neural networks is associated with the fact that convolution layers are equivariant with respect to the action of the group of translations, which makes these layers efficient for detecting features which naturally have this property. It is also easy to show that linear convolution operators are the only linear operators which are equivariant with respect to the natural action of the translation group on feature maps.

We introduce the following notations to describe the action of the translation group: We consider a grayscale image as a scalar-valued function $\varphi(i, j)$ defined

on $\mathbb{Z}^2$ and an element of the group of translations as a vector (u, v) in $\mathbb{Z}^2$. The natural action T of the group of translations on an image can be described by the formula

$$T_{u,v}(\varphi)(i, j) = \varphi(i - u, j - v). \tag{5.1}$$

A model layer L is called equivariant with respect to translations if it satisfies

$$L(T_{u,v}\varphi) = T_{u,v}(L(\varphi)). \tag{5.2}$$

Let's now consider a localization model M which takes as input an image $\varphi(i, j)$ showing one object and produces as output the coordinates of the object present in this image. Such a model does not produce a feature map, so that the previous definition of translation equivariance cannot be used for this model. We remark however that the group of translations acts naturally on $\mathbb{Z}^2$ by the action $T'_{u,v}(i, j) = i+u, j+v$, and that the model M should have the equivariance property

$$M(T_{u,v}\varphi) = T'_{u,v}(M(\varphi)). \tag{5.3}$$

Indeed, if the complete image is translated by a vector (u, v), then the object present in this image is also translated, so that the associated coordinates have to be shifted according to the vector (u, v).

It is not difficult to see that in the same way that convolutional operators are the only linear operators equivariant with respect to translations, it is also possible to fully describe which elementary operators follow this specific equivariance property. We first remark however that we have to restrict the space of possible input maps φ: if φ is a constant function, it does not change under the action of the translation group, so that the equivariance property 5.3 cannot be satisfied with such a function. We then suppose that φ satisfies $\sum_p \varphi(p) = 1$ and consider that the domain of the operator M is the corresponding affine space $\mathcal{A}$. We also replace the linearity condition by an the following affinity condition:

For all $\alpha_i \in \mathbb{R}, \varphi_i \in \mathcal{A}$ so that $\sum_i \alpha_i = 1$, we have $M(\sum_i \alpha_i \varphi_i) = \sum_i \alpha_i M(\varphi_i)$.

We then have the following proposition:

Proposition 5.4.1. *An affine operator M which satisfies the equivariance property 5.3 has to be of the form*

$$M(\varphi) = C + \sum_{p \in \mathbb{Z}^2} \varphi(p)p \tag{5.4}$$

for some constant C in $\mathbb{R}^2$.

Proof: We write the input map φ as a sum of spatially shifted versions of the function $\delta \in \mathcal{A}$ satisfying $\delta(p) = 1$ for $p = (0, 0)$ and $\delta(p) = 0$ for $p \neq (0, 0)$:

$$\varphi(p) = \sum_{q \in \mathbb{Z}^2} \varphi(q)\delta(p - q)\,. \tag{5.5}$$

We then use the the affine property of M and equivariance property 5.3:

$$M(\varphi) = M(\sum_q \varphi(q)\delta(p - q)) \tag{5.6}$$

$$= \sum_q \varphi(q) M(\delta(p-q)) = \sum_q \varphi(q)(M(\delta) + q) \tag{5.7}$$

$$= (\sum_q \varphi(q)) M(\delta) + \sum_q \varphi(q) q \tag{5.8}$$

$$= M(\delta) + \sum_q \varphi(q) q, \tag{5.9}$$

which proves the proposition since $M(\delta)$ is a constant.

The proposition 5.4.1 can be interpreted as stating that in order to get an equivariant localization operator, the most straightforward method is to build a normalized attention map φ from the input image and compute the coordinates of the detected object using an attention mechanism with φ as attention map and pixel coordinates as target values. One remark that it is precisely what the soft-argmax operator is doing: It takes an unnormalized scalar map ϕ as input, normalizes it using a softmax operator, and then perform localization using the same formula as in 5.4.1:

$$\begin{aligned} \text{soft-argmax}(\phi) &= \sum_{p \in \mathbb{Z}^2} \text{softmax}(\phi)(p) p \\ &= \sum_{p \in \mathbb{Z}^2} \frac{e^{\phi(p)}}{\sum_{q \in \mathbb{Z}^2} e^{\phi(q)}} p \end{aligned} \tag{5.10}$$

This operation is called soft-argmax because it allows to compute in a differentiable way an estimate of the coordinates of the maximum of the input map ϕ. Using soft-argmax then appears to be the most natural way to get an equivariant localization operator.

5.5 Related work

Unsupervised object detection and segmentation. Unsupervised object detection and segmentation models are generally reconstruction models: They try to reconstruct the input image using a specific image rendering process which induces the required object-centric structure. In order to ensure that objects are properly detected, various objectness priors have been defined and implemented:

- pixel similarity priors. Some models consider the task of object segmentation as a clustering problem, which can be addressed using deterministic [Hwang et al., 2019, Locatello et al., 2020] or probabilistic [Engelcke et al., 2021, Greff et al., 2016, Van Steenkiste et al., 2018] methods: If the feature vectors associated to two different pixels of an image are very similar, then it is considered that these pixels should both belong to the same object or to the background.

- independence priors. Some models assume that the images are sampled from a distribution which follows a probabilistic model featuring some independence priors between objects and the background, and use variational [Greff et al., 2019, Engelcke et al., 2020] or adversarial [Chen et al., 2019b, Bielski and Favaro, 2019] methods to learn these distributions.

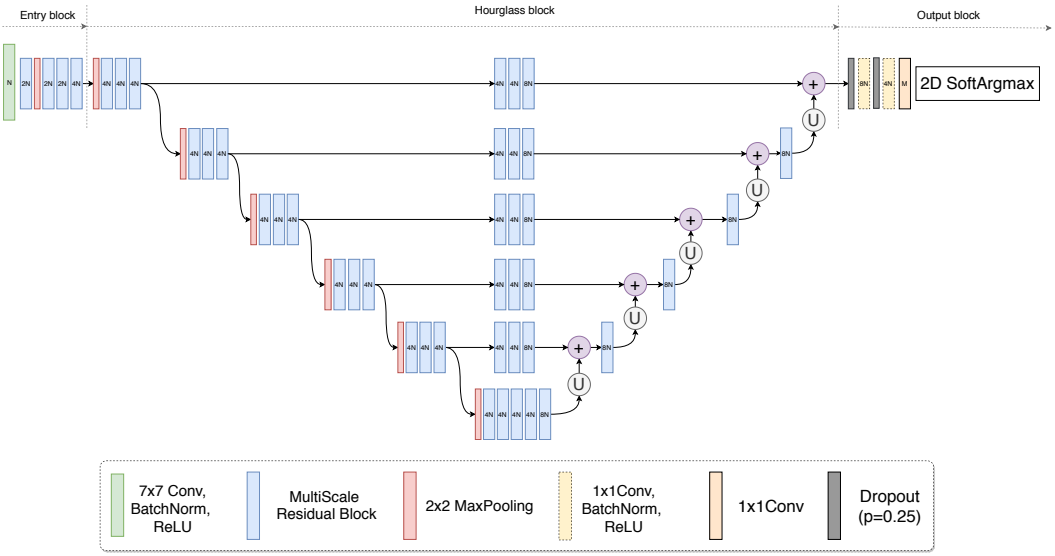


Figure 5.1: Overview of the KNEEL model [Tiulpin et al., 2019] , which uses a U-net and Soft argmax for anatomical landmark localization Source: [Tiulpin et al., 2019]

- disentanglement of appearance and location. Foreground objects appearing in the scenes of a given dataset can have similar shapes and appearances but very different scales and locations. Object discovery is performed by disentangling the object appearance generation process, which is performed by a convolutional glimpse generator [Ali Eslami et al., 2016, Kosiorek et al., 2018, Crawford and Pineau, 2019, Stelzner et al., 2019, Jiang et al., 2020, Jiang and Ahn, 2020] or a learned dictionary [Monnier et al., 2021, Smirnov et al., 2021], from the translation and scaling of the objects appearing in a scene, which is usually done by including a spatial transformer network [Jaderberg et al., 2015] in the model. The model described in this chapter belongs to this category and uses an convolutional glimpse generator.

Object detection and segmentation without spatial prior. State-of-the-art supervised detection and segmentation models usually rely on predefined reference anchors or center points which are spatially organized according to a periodic grid structure. The use of periodic grids has also been proposed for unsupervised object detection [Lin et al., 2020, Jiang et al., 2020, Jiang and Ahn, 2020, Smirnov et al., 2021]. Alternative detection methods relying on heatmaps produced by a U-net [Ronneberger et al., 2015] or stacked U-nets [Newell et al., 2016] networks, which predict for each pixel the probability of presence of one object on this pixel have been implemented in the supervised setting [Law and Deng, 2020, Duan et al., 2019].

For some specific applications such as human pose estimation or anatomical landmark localization [Tiulpin et al., 2019], some supervised models predict one heatmap per object. The use of soft-argmax for converting heatmaps to object coordinates has been implemented in the supervised [Sun et al., 2018, Luvizon et al., 2019, Chandran et al., 2020], semi-supervised [Honari et al.,

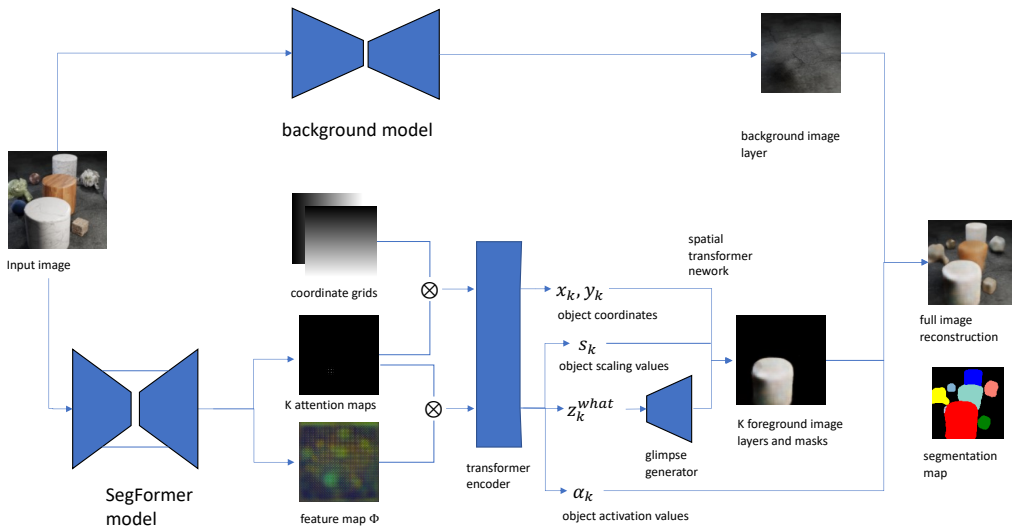


Figure 5.2: Overview of proposed model. A high resolution feature map generator (Segformer model) is trained to produce a high resolution feature map Φ and K scalar attention maps (one per object query). These maps are used to predict the coordinates and scales of the detected objects and the associated feature vectors, which are refined by a transformer encoder and then used as inputs to a glimpse generator and a spatial transformer network to produce K object image layers and masks. A convolutional autoencoder is in charge of background reconstruction.

2018] and unsupervised settings [Goroshin et al., 2015, Finn et al., 2016] to localize important features, but has never been proposed for unsupervised object detection or segmentation.

More recently, transformer-based [Vaswani et al., 2017] models using object [Carion et al., 2020, Zhu et al., 2021, Dong et al., 2021] or mask [Cheng et al., 2021, Cheng et al., 2022] queries have been proposed which not not rely explicitly on a spatial grid. These models show that transformers are efficient in the supervised setting to avoid multiple detections of the same object.

5.6 Description of proposed model

5.6.1 Model architecture

The overall architecture of the model is described in Fig 5.2.

The proposed model is composed of a a foreground model and a background model.

The background model is a deterministic convolutional autoencoder: We rely again on the classical assumption [Wright et al., 2009] that background images lie on a low-dimensional manifold, and use the autoencoder to learn this manifold.

The foreground model is also deterministic and associates to each object in the scene an appearance vector z^{what} which is used to produce a glimpse of the object, which is then scaled and translated at the right position on the image using a spatial transformer network.

The foreground encoding and reconstruction process can be described as follows: First, a high resolution feature map generator takes a color image of size $h \times w$ as input and produces a high resolution feature map Φ of dimension d_Φ and several scalar attention logit maps $A_1, ..., A_K$. We will use in this chapter the transformer-based Segformer model [Xie et al., 2021], which produces feature maps of size $h^* \times w^* = h/4 \times w/4$. The hyperparameter K is set to the maximum number of objects on a scene in the dataset. The scalar attention logit maps $A_1, ..., A_K$ are transformed into a normalized attention maps $\mathcal{A}_1, ..., \mathcal{A}_K$ using a softmax operator:

$$\mathcal{A}_k(i,j) = \frac{e^{A_k(i,j)}}{\sum_{i',j'} e^{A_k(i',j')}} \,. \tag{5.11}$$

We normalize the pixel indices (i,j) from the range $[1,..,w^*]$ and $[1,..,h^*]$ to the range $[-1,1]$ required by spatial transformer networks using the formulas

$$x(i) = 2\frac{i-1}{w^*-1} - 1 \tag{5.12}$$

$$y(j) = 2\frac{j-1}{h^*-1} - 1, \tag{5.13}$$

and predict initial estimates x_k^0, y_k^0 of the coordinates of the detected objects as the center of mass of the attention maps $\mathcal{A}_k$:

$$x_k^0 = \sum_{i=1,j=1}^{w^*,h^*} \mathcal{A}_k(i,j)x(i) \tag{5.14}$$

$$y_k^0 = \sum_{i=1,j=1}^{w^*,h^*} \mathcal{A}_k(i,j)y(j) \,. \tag{5.15}$$

We also build K object query feature vectors $\phi_1^0, ..., \phi_K^0$ of dimension d_Φ using the same attention maps $\mathcal{A}_1, .., \mathcal{A}_K$ as weights and the feature map Φ as target values:

$$\phi_k^0 = \sum_{i=1,j=1}^{w^*,h^*} \mathcal{A}_k(i,j)\Phi(i,j) \,. \tag{5.16}$$

A transformer encoder then takes the K triplets $(\phi_k^0, x_k^0, y_k^0)_{1 \leq k \leq K}$ as inputs and produces a refined version $(\phi_k, x_k, y_k)_{1 \leq k \leq K}$ taking into account possible detection redundancies and object occlusions. More precisely, we use a learned linear embedding to increase the dimension of the triplets (ϕ_k^0, x_k^0, y_k^0) from $d_\Phi + 2$ to the input dimension d_T of the transformer encoder, and a learned linear projection to reduce the dimension of the outputs of the transformer encoder from d_T back to $d_\Phi + 2$. The transformer encoder does not take any positional encoding as input, considering that the transformation which has to be performed should not depend on the ordering of the detections.

We force the final values of x_k and y_k to stay in the range $[-1,1]$ using clamping. Each transformed feature vector ϕ_k is then split in three terms: $\phi_k = (s_k, \alpha_k, z_k^{what})$.

- The first term s_k is an inverse scaling factor. It is a scalar if objects in the dataset have widths and heights which are similar (isotropic scaling), or a pair of scalars s_k^x, x_k^y if this is not the case (anisotropic scaling). We force the values of s_k to stay within a fixed range using a sigmoid function. The maximum value of this range ensures that a non-zero gradient will be available. The minimum value is set higher than 1 to make sure that the glimpse generator will not try to generate a full image layer.

- The second term is a scalar which is assumed to predict the activation level α_k of the object, which will be used to predict whether it is visible or not. We force this activation value to be positive using an exponential map.

- The remaining coordinates form a vector z_k^{what} which codes for the appearance of the object.

We then use a convolutional glimpse generator to build a color image o_k of the associated object together with the associated scalar mask m_k, using z_k^{what} as input. These images and masks are translated to the positions (x_k, y_k) and scaled according to the inverse scaling factor s_k using a spatial transformer network. We note L_k and M_k for $k \in \{1, .., K\}$ the corresponding object image layers and masks, and L_0 the background image produced by the background model, so that we have a total of $K + 1$ image layers.

We now have to decide for each pixel whether this pixel should show the background layer or one of the K object layers. In order to do this in a differentiable way, we multiply the predicted object masks M_k with the associated object activation levels α_k, and normalize the results to get one normalized weights distribution $(w_k)_{0 \leq k \leq K}$ per pixel:

$$w_k(i,j) = \frac{\alpha_k M_k(i,j)}{\sum_{k' \in 0..K} \alpha_{k'} M_{k'}(i,j)}, \tag{5.17}$$

considering that the mask M_0 associated to the background is set to 1 everywhere and that it has a fixed learned activation factor α_0.

The final reconstructed image $\hat{X}$ is then equal to the weighted sum of the various image layers using the weights w_k:

$$\hat{X}(i,j) = \sum_{k=0}^{K} w_k(i,j) L_k(i,j) \tag{5.18}$$

During inference the segmentation map is built by assigning to each pixel the layer index $k \in \{0, .., K\}$ for which $w_k(i,j)$ is the maximum. The background model is not needed to get the segmentation maps during inference.

5.6.2 Model training

Loss function

In order to train the proposed model, we use a main reconstruction loss function and an auxiliary loss:

Reconstruction loss. The local L_1 Reconstruction error associated to the pixel (i, j) is

$$l_{i,j} = \sum_{c=1}^{3} |\hat{x}_{c,i,j} - x_{c,i,j}|, \tag{5.19}$$

where $x_{c,i,j}$ and $\hat{x}_{c,i,j}$ are the values of the color channel c at the position (i, j) in the input image and reconstructed image.

The reconstruction loss is defined as the mean square of this reconstruction error.

$$\mathcal{L}_{rec} = \frac{1}{hw} \sum_{i=1,j=1}^{w,h} l_{i,j}^2 \tag{5.20}$$

Pixel entropy loss. For a given pixel (i, j), we expect the distribution of the weights $w_0(i, j), .., w_K(i, j)$ to be one-hot, because we assume that the objects are opaque. We observe that a discrete distribution is one-hot if and only if it has a zero entropy, so that minimizing the entropy of this distribution would be a reasonable way to enforce a stick-breaking process. Considering however that the entropy function has a singular gradient near one-hot distributions, we use the square of the entropy function to build the loss function. We then define the pixel entropy loss as

$$\mathcal{L}_{pixel} = \frac{1}{hw} \sum_{i=1,j=1}^{w,h} (\sum_{k=0}^{K} w_k(i, j) \log(w_k(i, j) + \epsilon))^2, \tag{5.21}$$

where $\epsilon = 10^{-20}$ is introduced to avoid any numerical issue with the logarithm function.

This auxiliary loss is weighted using the weight λ_{pixel} before being added to the reconstruction loss.

During our experiments, we observed that the pixel entropy loss could prevent a successful initialization of the localization process during the beginning of the training. As a consequence, we smoothly activate this auxiliary loss during initialization using a quadratic warmup of the weight.

The full loss function is then equal to

$$\mathcal{L} = \mathcal{L}_{rec} + \min(1, \frac{step}{N_{pixel}})^2 \lambda_{pixel} \mathcal{L}_{pixel}, \tag{5.22}$$

where *step* is the current training iteration index and N_{pixel} is a fixed hyperparameter.

Curriculum training

The interaction between the background reconstruction model and the foreground model during training is a very challenging issue, because of the competition between them to reconstruct the image. We handle this problem as in [Jiang and Ahn, 2020] by implementing curriculum training. We will then evaluate two methods to train the proposed model:

- baseline training (BT) : The background and foreground models are initialized randomly and trained simultaneously.

- curriculum training (CT): The training of the model is split in three phases:
    1. The background model is pretrained alone, using the methodology and robust loss function described in chapter 4.
    2. The weights of the background model are then frozen and the foreground model is trained using the frozen background model.
    3. The background and foreground models are then fine-tuned simultaneously.

5.7 Experimental results

5.7.1 Evaluation on public benchmarks

We perform a quantitative evaluation of the proposed model on the following datasets: CLEVRTEX [Karazija et al., 2021], CLEVR [Johnson et al., 2017], ShapeStacks [Groth et al., 2018] and ObjectsRoom [Kabra et al., 2019].

We implement on ShapeStacks, ObjectsRoom and CLEVR the same preprocessing as in [Engelcke et al., 2021].

We use the same hyperparameter values on these datasets, except for the hyperparameter K related to the number of object queries, which is set to the maximum number of objects in each dataset (i.e. 3 on ObjectsRoom, 6 on ShapeStacks and 10 on CLEVRTEX and CLEVR). We use isotropic scaling on CLEVR and ShapeStacks and anisotropic scaling on the other datasets.

We use the versions B3 of the Segformer model, and rely on the Hugging Face implementation of this model, with pretrained weights on ImageNet-1k for the hierarchical transformer backbone, but random initialization for the MLP decoder which is used as a feature map generator. We use the standard Pytorch implementation of the transformer encoder. The architecture of the backgroud model autoencoder is the same as in chapter 4. The glimpse generator is a sequence of transpose convolution layers, group normalization [Wu and He, 2020] layers and CELU [Barron, 2017] non-linearities, and is described in the appendix.

We use Adam as optimizer. The training process includes a quadratic warmup of the learning rate since the model contains a transformer encoder. We also decrease the learning rate by a factor of 10 when the number of training steps reaches 90% of the total number of training steps. The total number of training steps of the baseline training (BT) scenario is 125,000. In the curriculum training (CT) scenario, the number of training steps for background model pretraining (phase 1) is 500,000 on CLEVRTEX, ShapeStacks and ObjectsRoom, but 2500 on CLEVR, which shows a fixed background, as recommended in chapter 4. The number of training steps of phase 2 (training with frozen pretrained background model) is 30,000, and the number of training steps of the final fine-tuning phase (phase 3) is 95,000.

Full implementation details and hyperparameter values are provided in the supplementary material, and the model code is available on the Github platform.

In order to compare our results with published models, we compute the following evaluation metrics: mean intersection over union (mIoU) and adjusted rand index restricted to foreground objects (ARI-FG). We also provide the mean

Table 5.1: Benchmark results on CLEVR and CLEVRTEX. Results are shown ($\pm\sigma$) calculated over 3 runs. Source: [Karazija et al., 2021]

Model	CLEVR ↑mIoU (%)	CLEVR ↑ARI-FG (%)	CLEVR ↓MSE	CLEVRTEX ↑mIoU (%)	CLEVRTEX ↑ARI-FG (%)	CLEVRTEX ↓ MSE
SPAIR [Crawford and Pineau, 2019]	65.95 ± 4.02	77.13 ± 1.92	55 ± 10	0.00 ± 0.00	0.00 ± 0.00	1101 ± 2
SPACE [Lin et al., 2020]	26.31 ± 12.93	22.75 ± 14.04	63 ± 3	9.14 ± 3.46	17.53 ± 4.13	298 ± 80
GNM [Jiang and Ahn, 2020]	59.92 ± 3.72	65.05 ± 4.19	43 ± 3	42.25 ± 0.18	53.37 ± 0.67	383 ± 2
MN [Smirnov et al., 2021]	56.81 ± 0.40	72.12 ± 0.64	75 ± 1	10.46 ± 0.10	38.31 ± 0.70	335 ± 1
DTI [Monnier et al., 2021]	48.74 ± 2.17	89.54 ± 1.44	77 ± 12	33.79 ± 1.30	79.90 ± 1.37	438 ± 22
Gen-V2 [Engelcke et al., 2021]	9.48 ± 0.55	57.90 ± 20.38	158 ± 2	7.93 ± 1.53	31.19 ± 12.41	315 ± 106
eMORL [Emami et al., 2021]	50.19 ± 22.56	93.25 ± 3.24	33 ± 8	12.58 ± 2.39	45.00 ± 7.77	318 ± 43
MONet [Burgess et al., 2019]	30.66 ± 14.87	54.47 ± 11.41	58 ± 12	19.78 ± 1.02	36.66 ± 0.87	146 ± 7
SA [Locatello et al., 2020]	36.61 ± 24.83	95.89 ± 2.37	23 ± 3	22.58 ± 2.07	62.40 ± 2.23	254 ± 8
IODINE [Greff et al., 2019]	45.14 ± 17.85	93.81 ± 0.76	44 ± 9	29.17 ± 0.75	59.52 ± 2.20	340 ± 3
AST-Seg-B3-BT	71.92 ± 32.94	76.05 ± 36.13	51 ± 63	57.30 ± 15.72	71.79 ± 22.88	152 ± 39
AST-Seg-B3-CT	**90.27 ± 0.20**	**98.26 ± 0.07**	**16 ± 1**	**79.58 ± 0.54**	**94.77 ± 0.51**	**139 ± 7**

square error (MSE) between the reconstructed image and the input image, which provides an estimate of the accuracy of the learnt representation. We use the same definitions and methodology as [Karazija et al., 2021] for these metrics. We provide the mean segmentation covering (defined in [Engelcke et al., 2020]) restricted to foreground objects (MSC-FG) on ObjectsRoom and ShapeStacks where mIoU baseline values are not available.

We call AST-Seg (Attention and Soft-argmax with Transformer using Segformer) the proposed model, and AST-Seg-B3-BT, AST-Seg-B3-CT respectively the models using a Segformer B3 feature map generator trained under the baseline training or curriculum training scenarios. Tables 5.1 and 5.2 provide the results obtained on these datasets with a comparison with published results.

The proposed model trained under the baseline training scenario gets better average results than existing models on the CLEVR and CLEVRTEX dataset, but shows a very high variance. For example, on the CLEVR dataset, the model may fall during training in a bad minimum where the background model tries to predict the foreground objects. Using curriculum training allows to avoid this issue, get stable results on all datasets, and obtain a very significant mIoU improvement on the most complex datasets CLEVR and CLEVRTEX.

Table 5.2: Benchmark results on ObjectsRoom and ShapeStacks. Source: [Engelcke et al., 2021].

Model	ObjectsRoom ↑ARI-FG (%)	ObjectsRoom ↑MSC-FG (%)	ObjectsRoom ↑mIoU (%)	ObjectsRoom ↓MSE	ShapeStacks ↑ARI-FG (%)	ShapeStacks ↑MSC-FG (%)	ShapeStacks ↑mIoU (%)	ShapeStacks ↓MSE
MONet-g [Burgess et al., 2019]	54 ± 0	33 ± 1	n/a	n/a	70 ± 4	57 ± 12	n/a	n/a
Gen-v2 [Engelcke et al., 2021]	84 ± 1	58 ± 3	n/a	n/a	**81 ± 0**	68 ± 1	n/a	n/a
SA [Locatello et al., 2020]	79 ± 2	64 ± 13	n/a	n/a	76 ± 1	70 ± 5	n/a	n/a
AST-Seg-B3-BT	74.96 ± 10.02	69.86 ± 10.13	74.50 ± 8.61	11.7 ± 2.1	73.77 ± 7.56	74.12 ± 8.63	70.18 ± 12.68	11.8 ± 7.0
AST-Seg-B3-CT	**87.23 ± 0.88**	**82.22 ± 0.96**	**85.02 ± 0.79**	**6.7 ± 0.9**	79.34 ± 0.73	**77.65 ± 1.3**	**78.84 ± 0.21**	**4.5 ± 0.2**

Following the methodology proposed in [Karazija et al., 2021], we also evaluated the generalization capability of a model trained on CLEVRTEX when applied to datasets containing out of distribution images showing unseen textures and shapes or camouflaged objects (OOD and CAMO datasets [Karazija et al., 2021]). The results of this evaluation are provided in Table 5.3 and show

that the proposed model generalizes well, although it is deterministic and does not use any specific regularization scheme.

The OOD generalization results are significant because we have seen that the background/foreground segmentation model described in chapter 4 is not robust to domain shift. More generally, performing foreground segmentation in the presence of unseen significant background changes has always been considered a very significant challenge. We however observe that the proposed model is robust to domain shift and able to get reasonable background/foreground segmentation results on images containing unseen backgrounds.

Table 5.3: Benchmark generalization results on CAMO, and OOD for a model trained on CLEVRTEX. Results are shown ($\pm\sigma$) calculated over 3 runs. Source: [Karazija et al., 2021]

Model	OOD			CAMO		
	↑mIoU (%)	↑ARI-FG (%)	↓MSE	↑mIoU (%)	↑ARI-FG (%)	↓MSE
SPAIR [Crawford and Pineau, 2019]	0.00 ± 0.00	0.00 ± 0.00	1166 ± 5	0.00 ± 0.00	0.00 ± 0.00	668 ± 3
SPACE [Lin et al., 2020]	6.87 ± 3.32	12.71 ± 3.44	387 ± 66	8.67 ± 3.50	10.55 ± 2.09	251 ± 61
GNM [Jiang and Ahn, 2020]	40.84 ± 0.30	48.43 ± 0.86	626 ± 5	17.56 ± 0.74	15.73 ± 0.89	353 ± 1
MN [Smirnov et al., 2021]	12.13 ± 0.19	37.29 ± 1.04	409 ± 3	8.79 ± 0.15	31.52 ± 0.87	265 ± 1
DTI [Monnier et al., 2021]	32.55 ± 1.08	73.67 ± 0.98	590 ± 4	27.54 ± 1.55	72.90 ± 1.89	377 ± 17
Gen-V2 [Engelcke et al., 2021]	8.74 ± 1.64	29.04 ± 11.23	539 ± 147	7.49 ± 1.67	29.60 ± 12.84	278 ± 75
eMORL [Emami et al., 2021]	13.17 ± 2.58	43.13 ± 9.28	471 ± 51	11.56 ± 2.09	42.34 ± 7.19	269 ± 31
MONet [Burgess et al., 2019]	19.30 ± 0.37	32.97 ± 1.00	**231 ± 7**	10.52 ± 0.38	12.44 ± 0.73	**112 ± 7**
SA [Locatello et al., 2020]	20.98 ± 1.59	58.45 ± 1.87	487 ± 16	19.83 ± 1.41	57.54 ± 1.01	215 ± 7
IODINE [Greff et al., 2019]	26.28 ± 0.85	53.20 ± 2.55	504 ± 3	17.52 ± 0.75	36.31 ± 2.57	315 ± 3
AST-Seg-B3-CT	**67.50 ± 0.75**	**83.14 ± 0.75**	832 ± 24	**73.07 ± 0.65**	**87.27 ± 3.78**	145 ± 6

Some segmentation prediction samples are provided in Fig 5.3. Other image samples are available in the appendix to this chapter.

The main limitation of the proposed model is the management of shadows, which may be considered by the model as separate objects or integrated to object segmentations.

5.7.2 Quality of learned object representations

Fig 5.5 shows examples of object glimpses produced by the glimpse generator on the CLEVRTEX dataset.

In order to check whether the object feature vectors z_{what} learned by the model can be useful for downstream tasks, we also provide in Fig. 5.6 t-SNE plots of the distribution of the vectors z_{what} associated to positive detections (non-zero segmentation masks) on the ObjectsRoom and CLEVRTEX datasets, which show that the learned object representations are smooth. The object representations obtained on the ObjectsRoom also appear to be meaningful, allowing to distinguish the various shapes of the objects appearing in this dataset.

5.7.3 Qualitative evaluation on real-world traffic videos

We have also tested the proposed model on real-world videos downloaded from free webcam sites available on the Internet. We selected three traffic webcams[1]

[1] https://www.youtube.com/watch?v=YByJ2h0T5JY, https://www.youtube.com/watch?v=BGCytWLOmyA, https://www.youtube.com/watch?v=DQe_EvBae_I

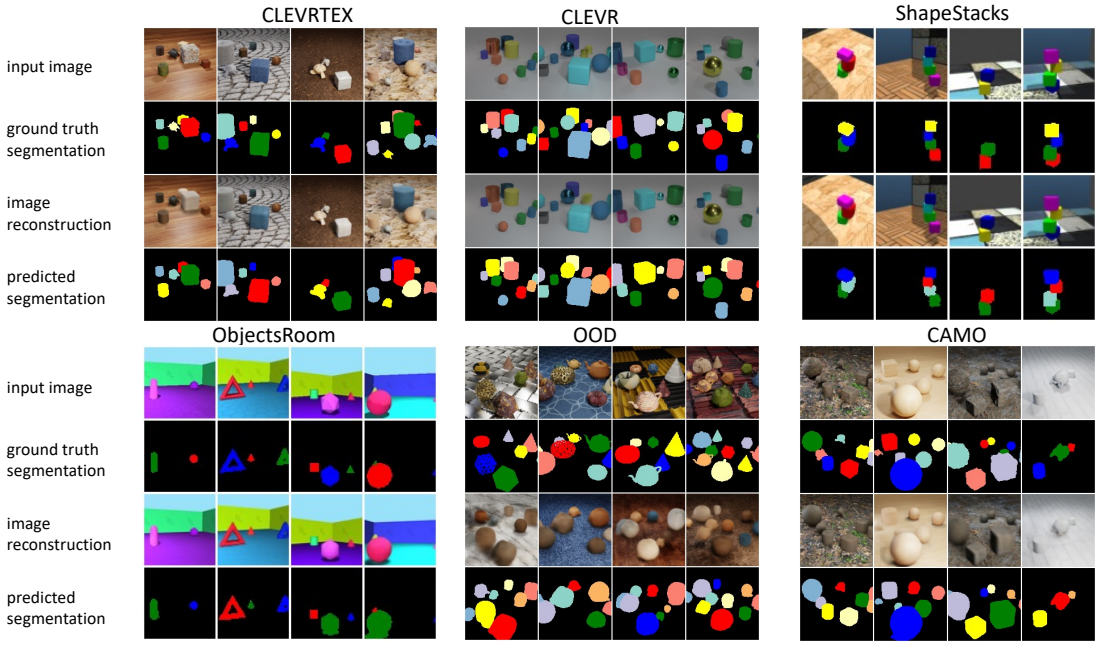


Figure 5.3: Examples of segmentation predictions on CLEVRTEX, CLEVR, ShapeStacks, ObjectsRoom, OOD and CAMO test datasets (Results on OOD and CAMO datasets are obtained using a model trained on CLEVRTEX only)

showing significant challenges such as high level of occlusion, object size diversity or presence of pedestrians. For each website, we downloaded 5 hours of videos at 5 frames per second, then extracted selected regions of interest which were resized to 200x320 images, and checked that these patches did not allow to identify any person, license plate or other personally identifiable information. Since some of these videos show fixed background objects such as traffic lights or posts which may occlude foreground objets, we adapted the model to take into account this issue by replacing the uniform learnt background activation α_0 by a pixel-wise learnt background activation $\alpha_0(i, j)$.

Examples of predicted segmentations are provided in Fig 5.7. Other samples are available in the appendix. Examples of generated glimpses are provided in Fig 5.8. The distribution of the associated z_{what} vectors is illustrated in Fig 5.9 and 5.10, and shows that the latent space is smooth and meaningful, allowing to separate cars from pedestrians and to distinguish cars according to their orientations.

5.7.4 Ablation study and additional experiments

We provide in Table 5.4 results obtained using various ablations or modifications on the model architecture or loss function, which show that:

- The model remains competitive if the transformer encoder is removed by setting $(\phi_k, x_k, y_k)_{1 \leq k \leq K} = (\phi_k^0, x_k^0, y_k^0)_{1 \leq k \leq K}$. The results on the ShapeStacks and ObjectsRoom datasets are even improved with this simplified

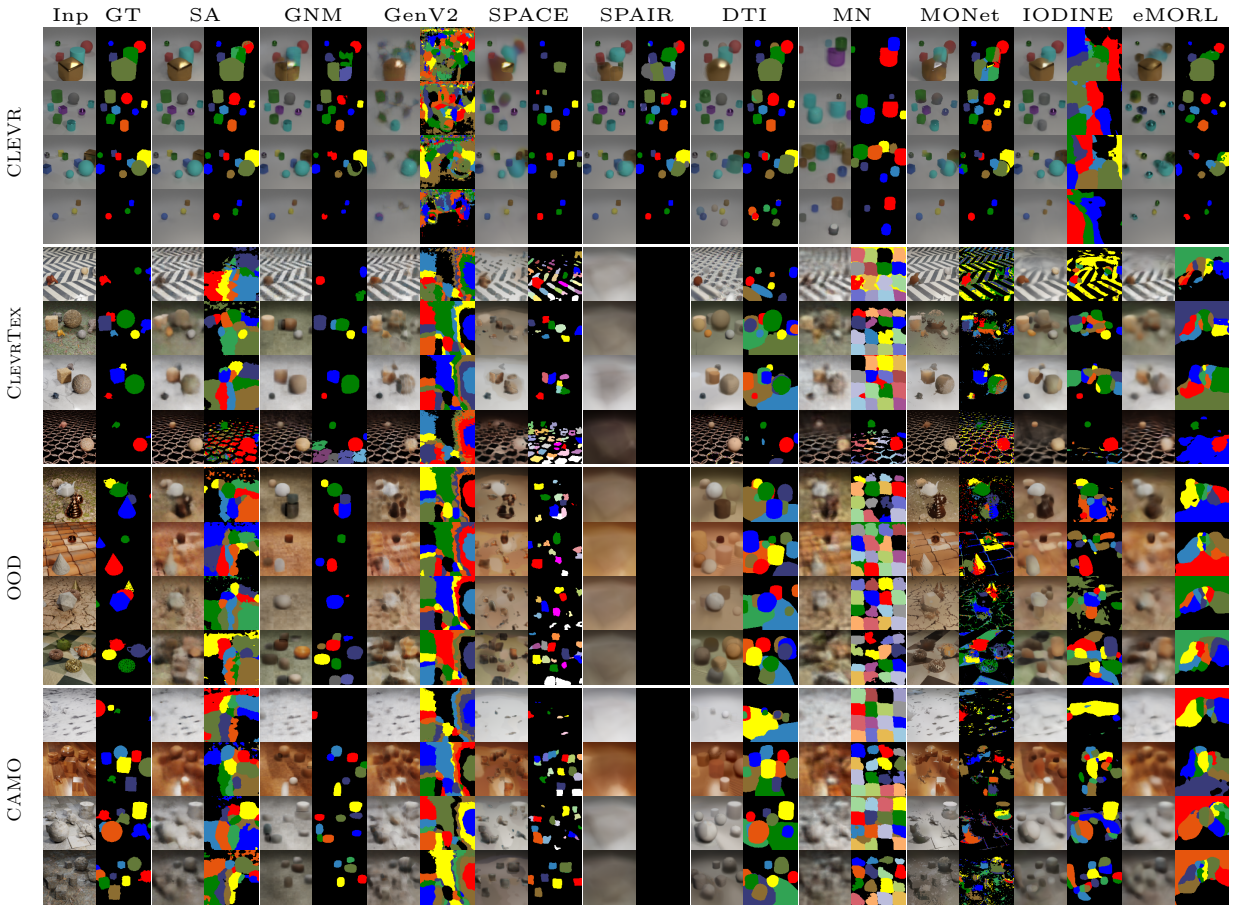


Figure 5.4: Examples of segmentation predictions on CLEVRTEX, CLEVR, OOD and CAMO test datasets obtained using other models (best viewed digitally). Source: [Karazija et al., 2021]

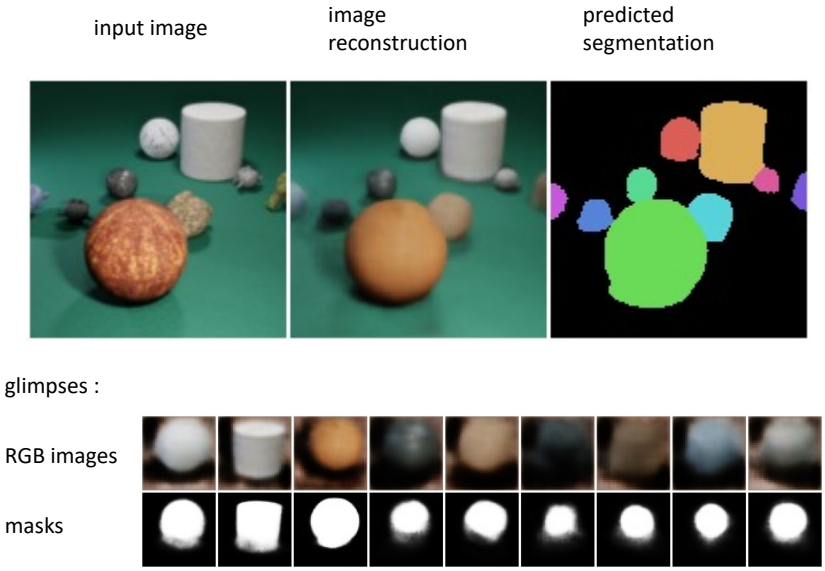


Figure 5.5: Examples of glimpses (RGB images and masks) generated by the glimpse generator on an image from the CLEVRTEX dataset

Table 5.4: Results of ablation study and additional experiments (results over 1 run, except for starred values, which are averages over 3 runs)

Dataset	CLEVRTEX		CLEVR		ShapeStacks		ObjectsRoom	
	mIoU	ARI-FG	mIoU	ARI-FG	mIoU	ARI-FG	mIoU	ARI-FG
full model AST-Seg-B3-CT (reference)	79.58*	94.77*	90.27*	98.26*	78.84*	79.34*	85.02*	87.23*
model without transformer encoder	75.69	94.41	77.16	93.09	82.99*	82.29*	85.51*	88.49*
K = 1 + maximum number of objects	79.11*	94.78*	91.03*	98.17*	78.87	80.05	82.90	86.45
K = 2 × maximum number of objects	62.10	89.96	90.56	98.29	54.88	65.16	66.78	78.58
using a Unet instead of Segformer feature generator	66.82	88.25	90.70	98.17	75.51	77.78	85.59	87.93
random initialization of Segformer backbone	61.74	80.22	88.94	97.77	62.73	68.40	77.71	79.23
training without pixel entropy loss	70.18	91.81	85.54	96.09	52.17	60.08	84.21	86.19
training using frozen pretrained background model	75.30	95.31	81.46	98.29	55.06	66.24	85.82	87.78
isotropic scaling	78.68	94.78					84.91	87.20
anisotropic scaling			87.21	98.53	45.47	36.43		

architecture, with a surprisingly strong improvement on the Shapestacks dataset, which shows the efficiency of the attention and soft-argmax mechanism. The transformer encoder is however necessary on the more complex CLEVR and CLEVRTEX datasets.

- Training with a number of slots slightly higher than the maximum number of objects does not lead to significant changes in the results. A more substantial increase of the number of slots however leads to poor results on scenes with complex textures due to an increasing fragmentation of the objects. This is very different from the situation observed on query-based

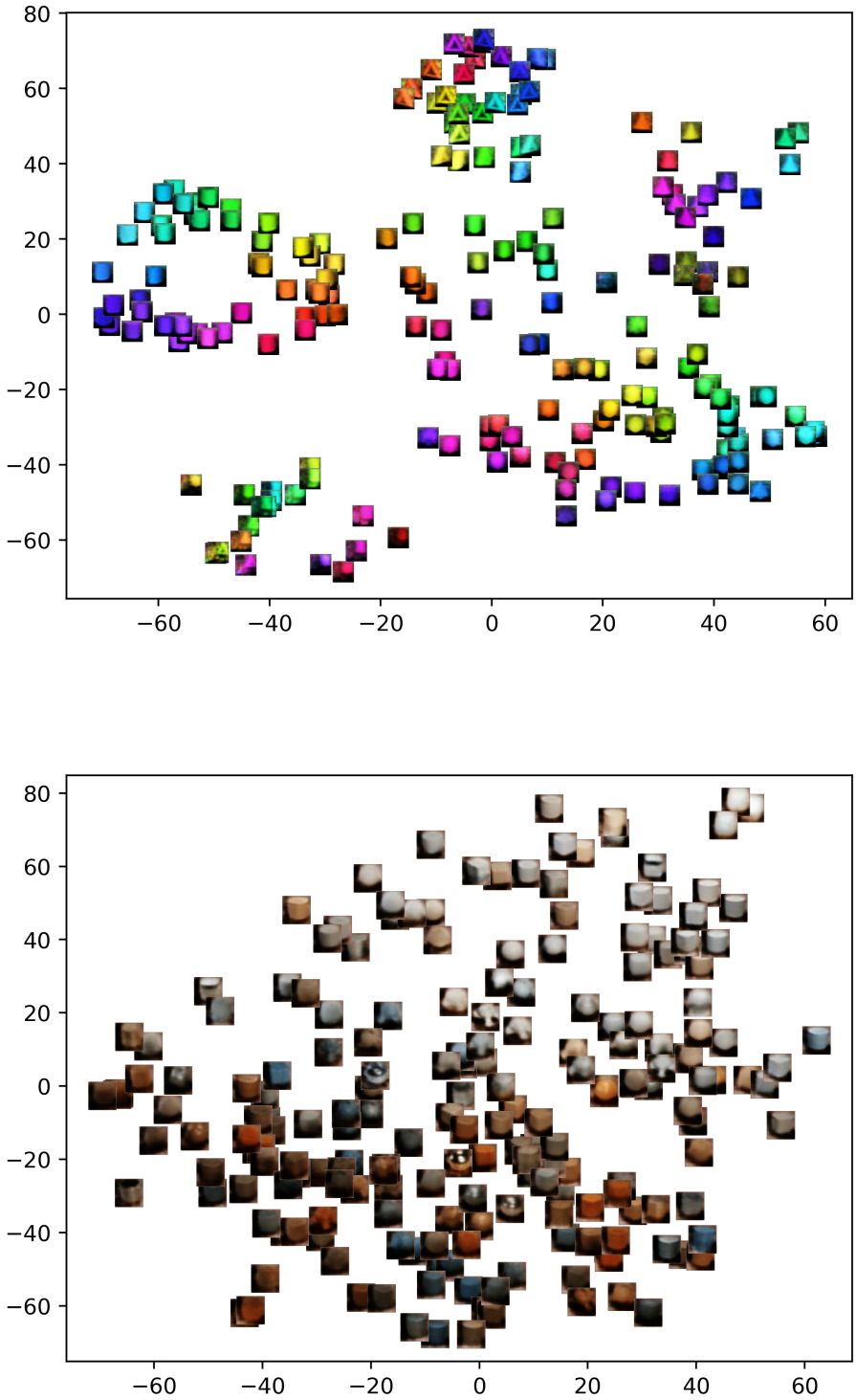


Figure 5.6: t-SNE plots of the distribution of the z_{what} vectors on the ObjectsRoom and CLEVRTEX datasets. Each z_{what} vector is represented by the associated RGB glimpse

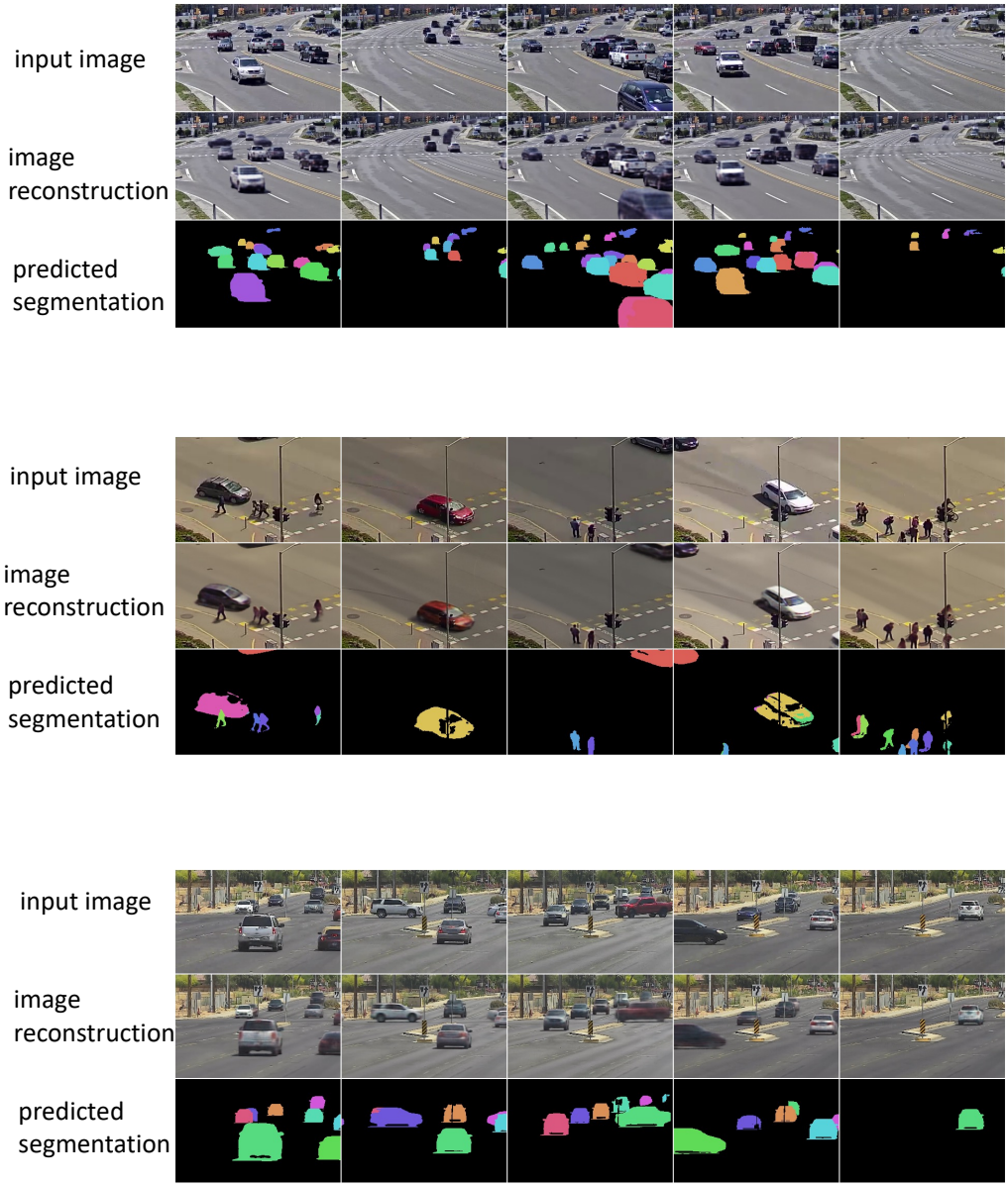
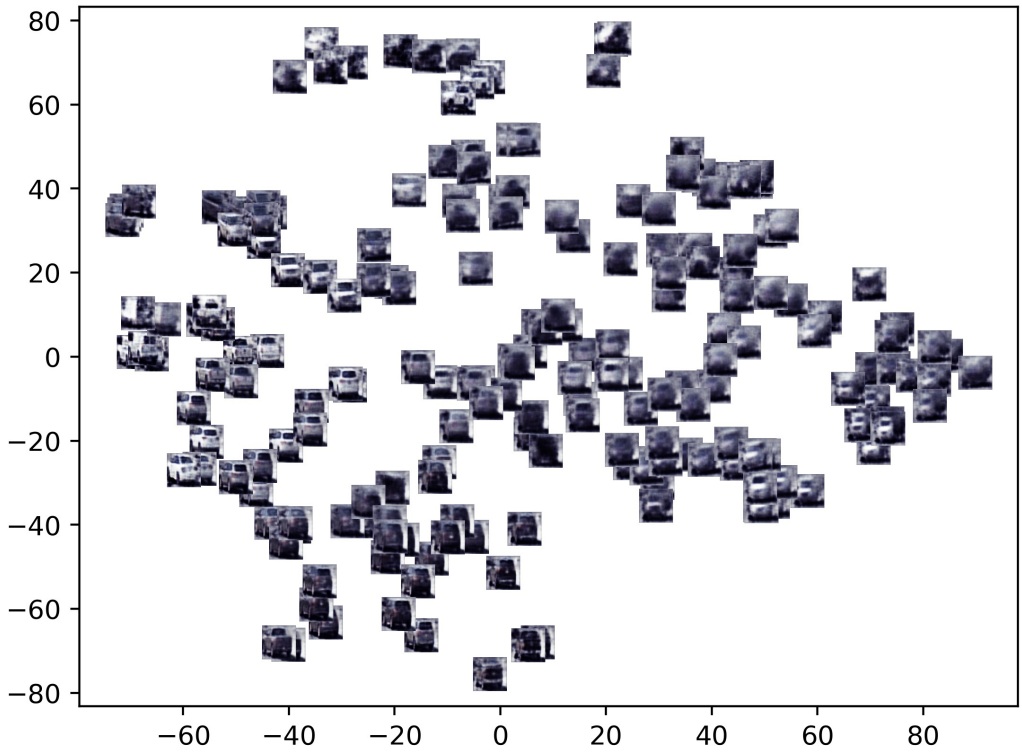


Figure 5.7: Examples of segmentation predictions on real-world traffic videos

Figure 5.8: Examples of object glimpses generated from real-world traffic videos and associated input images, reconstructed images and predicted segmentation maps



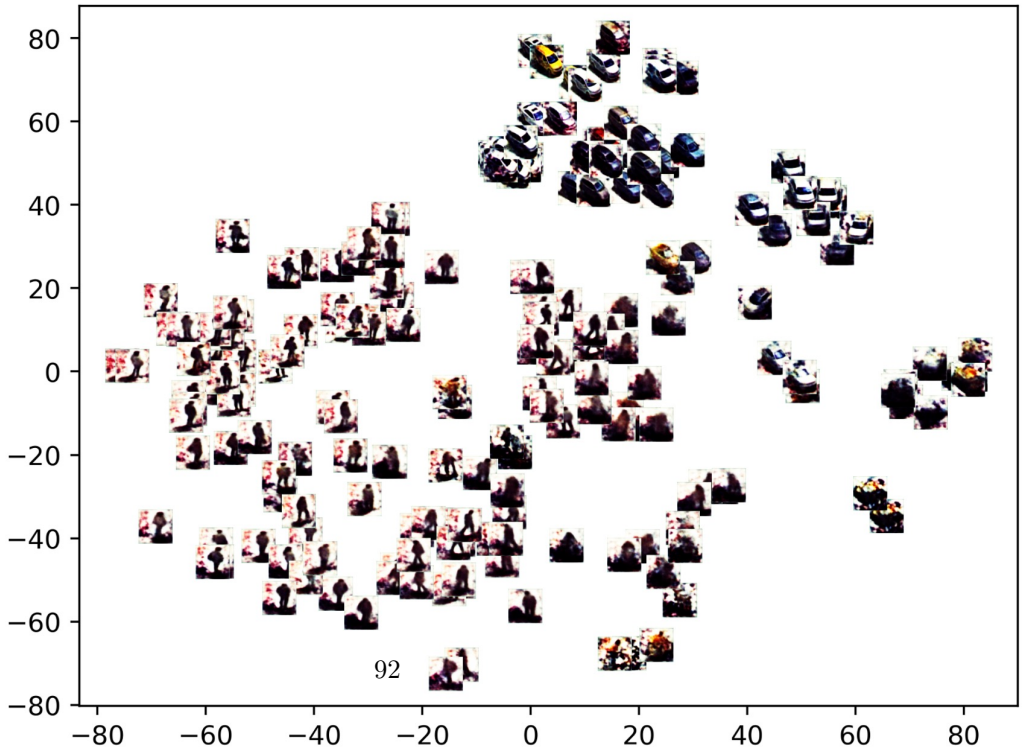


Figure 5.9: t-SNE plots of the distribution of the z_{what} vectors associated to positive detections on two real-world traffic videos

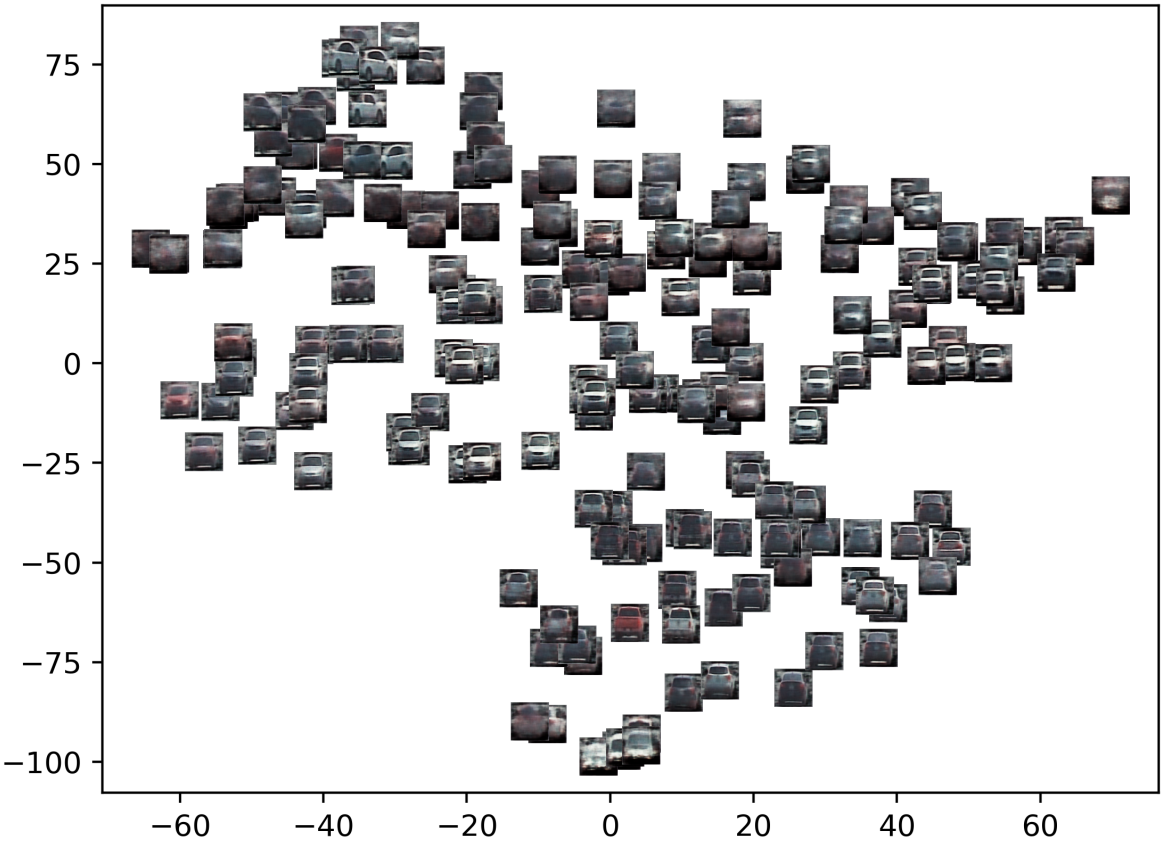


Figure 5.10: t-SNE plot of the distribution of the z_{what} vectors associated to positive detections on one real-world traffic video

supervised detection models like DETR, where the number of queries has to be very high compared to the number of objects.

- It is possible to replace the Segformer high resolution feature map generator with any other generator. The proposed model was originally designed with a custom Unet feature map generator, which gets similar results as the Segformer model on CLEVR, ShapeStacks and ObjectsRoom, but underperforms on the more complex CLEVRTEX dataset. The architecture of this Unet is described in the supplementary material.

- Using a pretrained backbone is necessary to get good performances with a Segformer feature map generator.

- We tested an alternative training scenario where the background model remains frozen during the complete training of the foreground model (125 000 iterations). The main advantage of this scenario is that it is significantly faster and requires less memory, since the backgrounds of the training images can be pre-computed and memorized. The accuracy of the results is however lower than the curriculum training scenario proposed in this chapter, except for the ObjectsRoom dataset.

- Switching between isotropic scaling and anisotropic scaling does not make much difference, except for the ShapeStacks dataset, where the proposed model can consider that each block tower is a single object if anisotropic scaling is enabled.

Table 5.5: Training computation time with one Nvidia RTX 3090 GPU (curriculum training)

Dataset	image size	background model pretraining (phase 1)		full model training (phase 2 & 3)	
		number of iterations	training time	number of iterations	training time
CLEVRTEX	128×128	500000	57 h 47 mn	125000	16 h 00 mn
CLEVR	128×128	2500	20 mn	125000	12 h 03 mn
ObjectsRoom	64×64	500000	14 h 57 mn	125000	6 h 31 mn
ShapeStacks	64×64	500000	14 h 20 mn	125000	6 h 22 mn

5.7.5 Computation time

All experiments have been performed using a Nvidia RTX 3090 GPU and a AMD 7402 EPYC CPU. Some training durations are provided in Table 5.5.

5.8 Conclusion of chapter 5

We have described in this chapter a new architecture for unsupervised object-centric representation learning and object detection and segmentation, which relies on attention and soft-argmax, and shown that this new architecture substantially improves upon the state of the art on existing benchmarks showing synthetic scenes with complex shapes and textures. We hope this work may help to extend the scope of structured object-centric representation learning from research to practical applications.

Table 5.6: Hyperparameter values

hyperparameter description	notation	value
Background model pretraining:		
batch size		128
learning rate		2.10^{-3}
number of background model training iterations:		
- datasets with fixed backgrounds (CLEVR)		2500
- datasets with complex backgrounds (CLEVRTEX, ShapeStacks,ObjectsRoom)		500000
Foreground model training:		
batch size		64
learning rate		4.10^{-5}
Adam β_1		0.90
Adam β_2		0.98
Adam ϵ		10^{-9}
number of foreground model training iterations		125000
number of steps of phase 2 (CT scenario)		30000
number of steps of learning rate warmup phase		5000
number of steps of pixel entropy loss weight warmup phase	N_{pixel}	10000
initial value of background activation before training	α_0	e^{11}
dimension of z_{what}	$d_{z_{what}}$	32
pixel entropy loss weight	λ_{pixel}	1.10^{-2}
minimum value of inverse scaling factor	s_{min}	1.3
maximum value of inverse scaling factor	s_{max}	24
dimension of inputs and outputs of transformer encoder	d_T	256
number of heads of transformer encoder layer		8
dimension of feedforward transformer layer		512
number of layers of transformer encoder		6

5.9 Appendix to chapter 5

5.9.1 Hyperparameter values

The hyperparameter values used for the proposed model are listed in Table 5.6.

5.9.2 Pseudo-code for objects encoder and decoder

The full encoding and rendering process is described in Algorithms 1 and 2.

5.9.3 Additional implementation details

The glimpse convolutional generator is described in Table 5.7.

Synthetic datasets and preprocessing codes were downloaded from the following public repositories:

- https://www.robots.ox.ac.uk/~vgg/data/clevrtex/
- https://ogroth.github.io/shapestacks/
- https://github.com/deepmind/multi_object_datasets
- https://github.com/applied-ai-lab/genesis.

Algorithm 1: Encoding

Input: input image $\mathbf{X}$

Output: object latents $\{\mathbf{z}_k^{what}, x_k, y_k, s_k, \alpha_k\}_{1 \geq k \geq K}$

`// feature and attention maps generation`

$(\Phi, A_1, .., A_K) = \text{Segformer}(\mathbf{X})$

for $k \leftarrow 1$ **to** K, $i \leftarrow 1$ **to** w^*, $j \leftarrow 1$ **to** h^* **do**

  $\mathcal{A}_k(i,j) = \text{Softmax}(A_k)(i,j) = \frac{e^{A_k(i,j)}}{\sum_{i,j} e^{A_k(i,j)}}$

end

`// computation of positions and feature vectors before transformer refinement`

for $i \leftarrow 1$ **to** w^*, $j \leftarrow 1$ **to** h^* **do**

  $x(i) = 2\frac{i-1}{w^*-1} - 1$; $y(j) = 2\frac{j-1}{h^*-1} - 1$

end

for $k \leftarrow 1$ **to** K,$i \leftarrow 1$ **to** w^*, $j \leftarrow 1$ **to** h^* **do**

  $x_k^0 = \sum_{i,j} x(i)\mathcal{A}_k(i,j)$; $y_k^0 = \sum_{i,j} y(j)\mathcal{A}_k(i,j)$

  $\phi_k^0 = \sum_{i,j} \Phi(i,j)\mathcal{A}_k(i,j)$

end

`// transformer refinement of positions and feature vectors`

$(x_k, y_k, \phi_k)_{1 \geq k \geq K} =$
$\text{LinearProjection}(\text{TransformerEncoder}(\text{LinearEmbedding}((x_k^0, y_k^0, \phi_k^0)_{1 \geq k \geq K})))$

`// latent computations`

for $k \leftarrow 1$ **to** K **do**

  $x_k = \text{clamp}(x_k, min = -1, max = 1)$; $y_k =$
  $\text{clamp}(y_k, min = -1, max = 1)$

  $(s_k, \alpha_k, z_k^{what}) = \phi_k$

  $s_k = s_{min} + (s_{max} - s_{min})\sigma(s_k)$

  $\alpha_k = e^{\alpha_k}$

end

Output: $\{\mathbf{z}_k^{what}, x_k, y_k, s_k, \alpha_k\}_{1 \geq k \geq K}$

Algorithm 2: Rendering

Input: object latents $\{\mathbf{z}_k^{what}, x_k, y_k, s_k, \alpha_k\}$, background image L_0, background mask $M_0 = 1$, learned background activation α_0 or $\alpha_0(i,j)$

Output: Image reconstruction $\hat{\mathbf{X}}$

```
// Obtain the object appearance o_k and segmentation mask m_k
```

for $k \leftarrow 1$ **to** K **do**

  $\mathbf{o}_k, \mathbf{m}_k = \text{GlimpseGenerator}(\mathbf{z}_k^{\text{what}})$

end

```
// translation and scaling using a spatial transformer
   network (STN)
```

for $k \leftarrow 1$ **to** K **do**

  $L_k = \text{STN}(\mathbf{o}_k, x_k, y_k, s_k)$

  $M_k = \text{STN}(\mathbf{m}_k, x_k, y_k, s_k)$

end

```
// occlusion computations
```

for $k \leftarrow 0$ **to** K **do**

  $w_k = \frac{\alpha_k M_k}{\sum_{i=0}^{K} \alpha_i M_i}$

end

```
// combination of image layers
```

$\hat{\mathbf{X}} = \sum_{k=0}^{K} w_k L_k;$

Output: $\hat{\mathbf{X}}$

Table 5.7: glimpse generator architecture

64x64 images

Layer	Size	Ch	Stride	Padding	Norm./Act.
Input	1	$d_{z_{what}}$			
Transp Conv 2×2	2	64	2	0	GroupNorm(4,64) /CELU
Transp Conv 4×4	4	32	2	1	GroupNorm(2,32)/CELU
Transp Conv4×4	8	16	2	1	GroupNorm(1,16)/CELU
Transp Conv 4×4	16	8	2	1	GroupNorm(1,8)/CELU
Transp Conv 4×4	32	4	2	1	
Sigmoid	32	4			

128x128 images

Layer	Size	Ch	Stride	Padding	Norm./Act.
Input	1	$d_{z_{what}}$			
Transp Conv 2×2	2	128	2	0	GroupNorm(8,128) /CELU
Transp Conv 4×4	4	64	2	1	GroupNorm(4,64)/CELU
Transp Conv4×4	8	32	2	1	GroupNorm(2,32)/CELU
Transp Conv 4×4	16	16	2	1	GroupNorm(1,16)/CELU
Transp Conv 4×4	32	8	2	1	GroupNorm(1,8)/CELU
Transp Conv 4×4	64	4	2	1	
Sigmoid	64	4			

Table 5.8: U-net architecture (ablation study)

Layer	Ch	Stride	Padding	Norm./Act.
Input	3			
Conv 3×3	80	1	1	BatchNorm /CELU
Downsample block	128			
Downsample block	192			
Downsample block	256			
Downsample block	256			
Downsample block	256			
Center block	256			
Upsample block	256			
Upsample block	256			
Upsample block	192			
Upsample block	128			
Upsample block	80			
Conv 3×3 with skip connection	d_Φ	1	1	BatchNorm /CELU
Residual Conv 3×3	d_Φ	1	1	
Conv 1×1	d_Φ	1	1	

The Segformer pretrained weights were downloaded from the following link: https://huggingface.co/nvidia/mit-b3

The architecture of the U-net implemented for the ablation study is described in Table 5.8. It contains a sequence of downsample blocks which output feature maps of decreasing sizes, a center block which takes as input the feature map produced by the last downsample block, and upsample blocks, which take as input both the output of the previous upsample or center block and the feature map of the same size produced by corresponding downsample block.

- A downsample block is composed of a convolutional layer with stride 2 and kernel size 4, with batch normalization and CELU, followed by a residual convolutional layer with stride 1 and kernel size 3 with batch normalization and CELU.
- The center block is composed of a convolutional layer with stride 1 and kernel size 3 with batch normalization and CELU.
- An upsample block is composed of a residual convolutional layer with stride 1 and kernel size 3 with batch normalization and CELU, followed by a transpose convolutional layer with stride 2 and kernel size 4, with batch normalization and CELU.

5.9.4 Additional image samples

Additional image samples are provided in Figs. 5.11 to 5.16 on pages 99–104

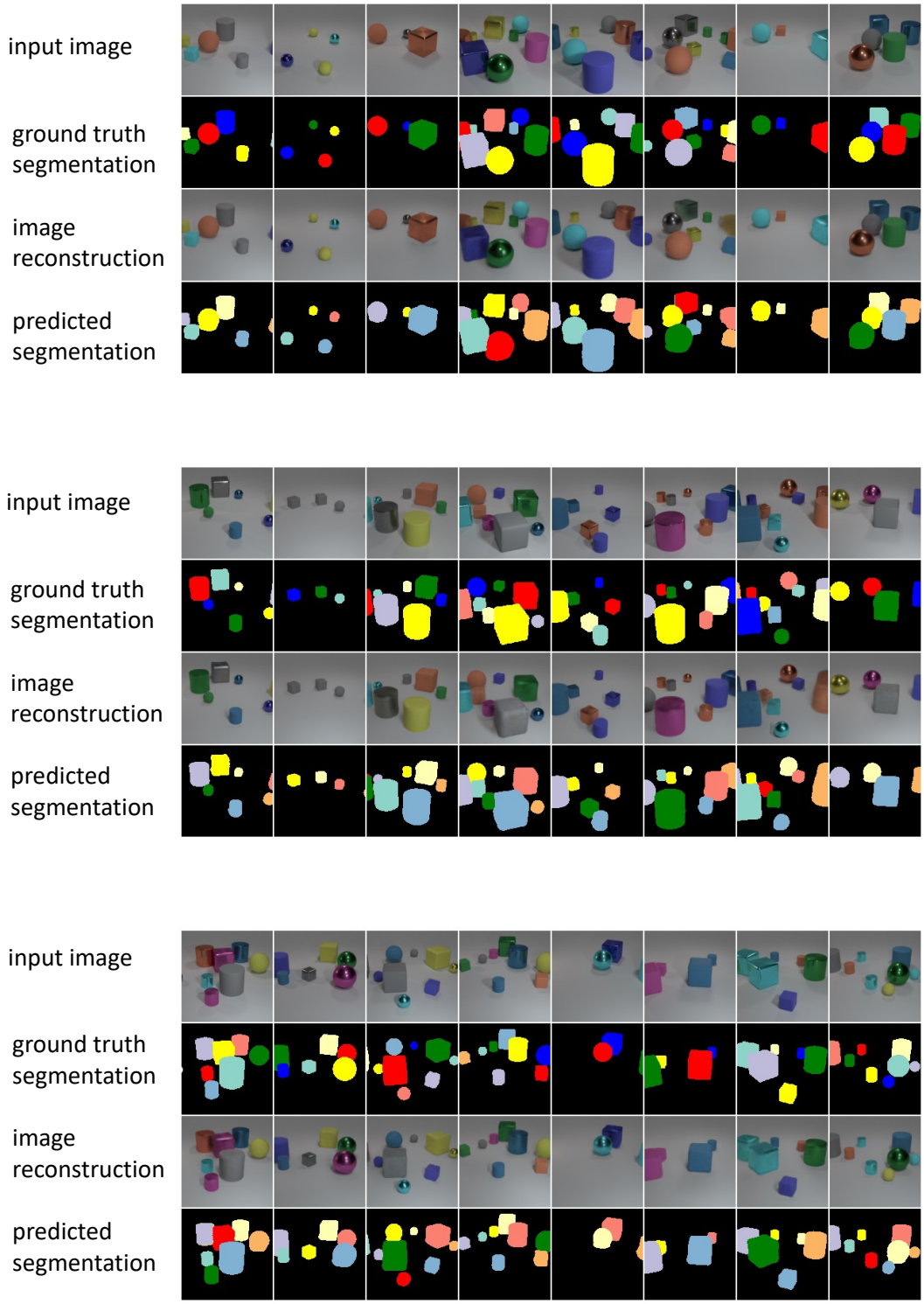


Figure 5.11: Examples of segmentation predictions on CLEVR test dataset

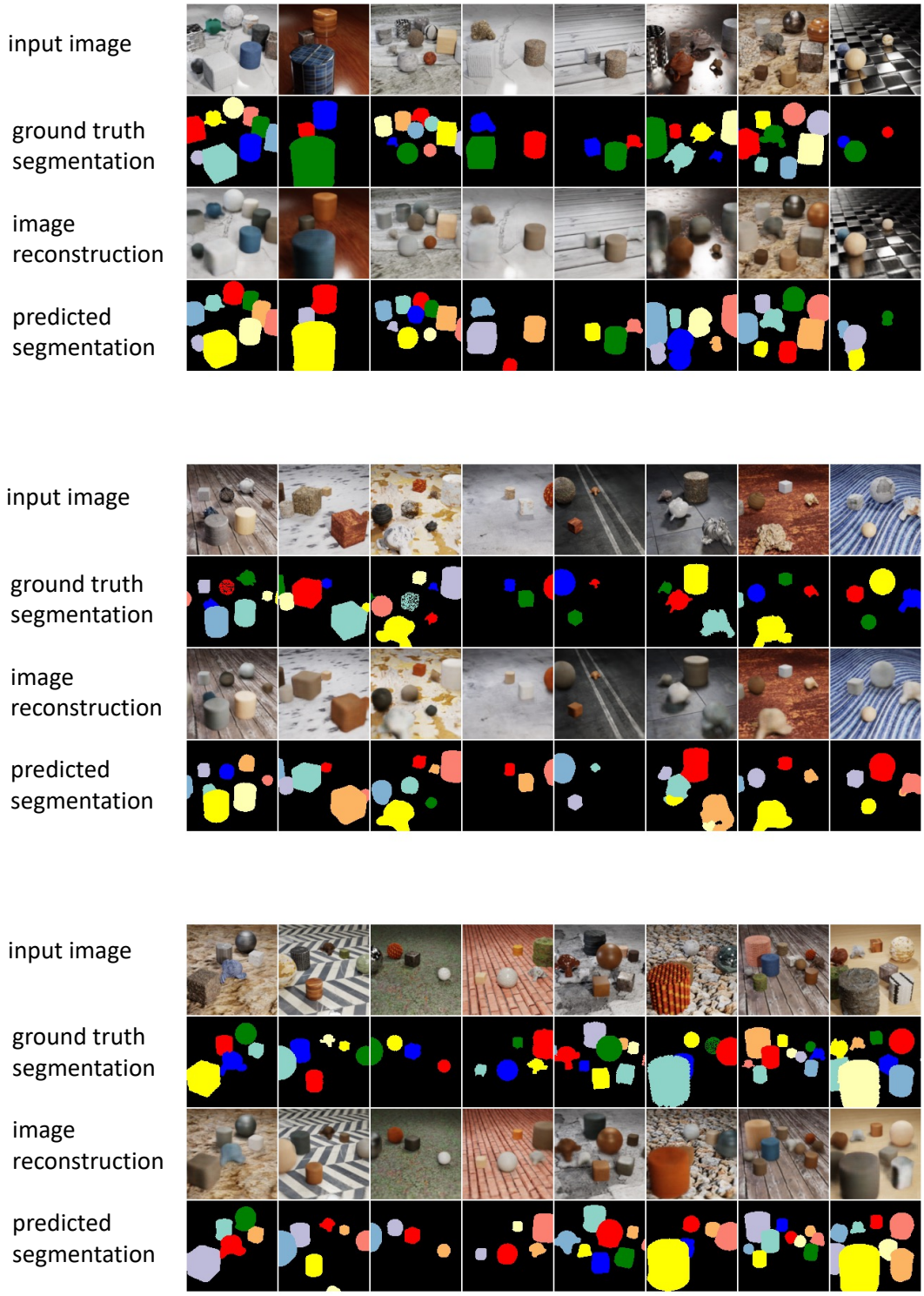


Figure 5.12: Examples of segmentation predictions on CLEVRTEX test dataset

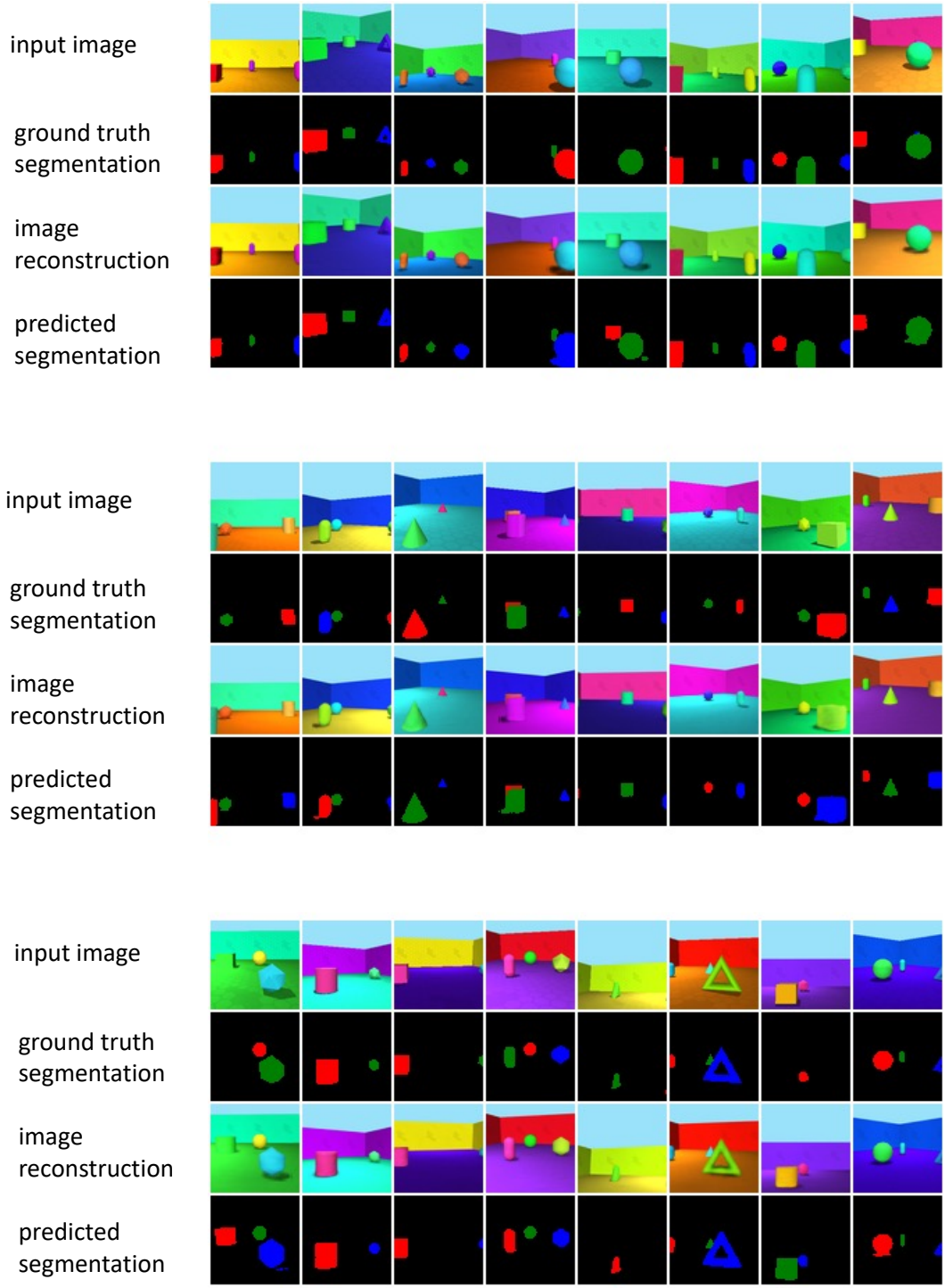


Figure 5.13: Examples of segmentation predictions on ObjectsRoom test dataset

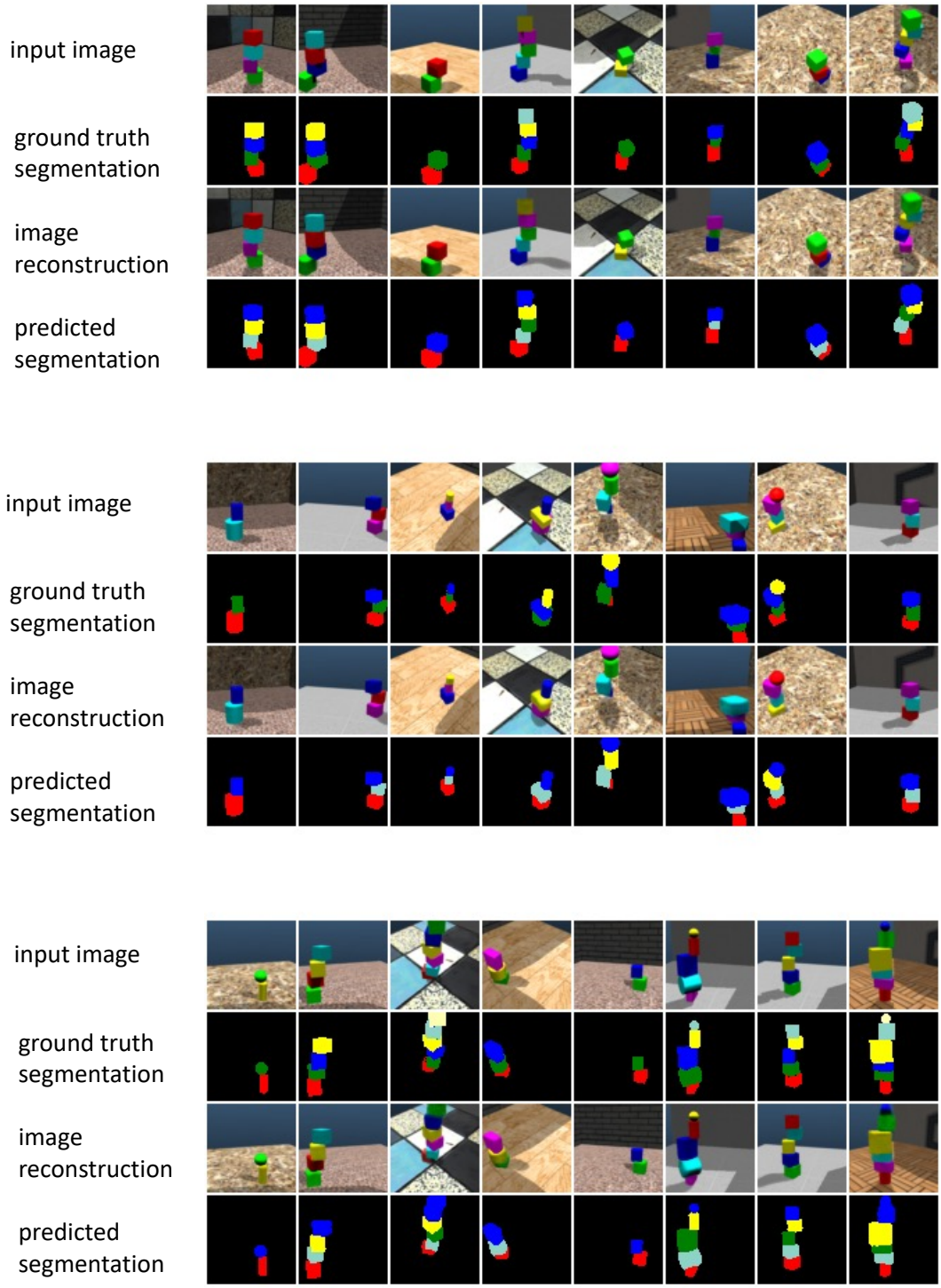


Figure 5.14: Examples of segmentation predictions on ShapeStacks test dataset (using a model without transformer)

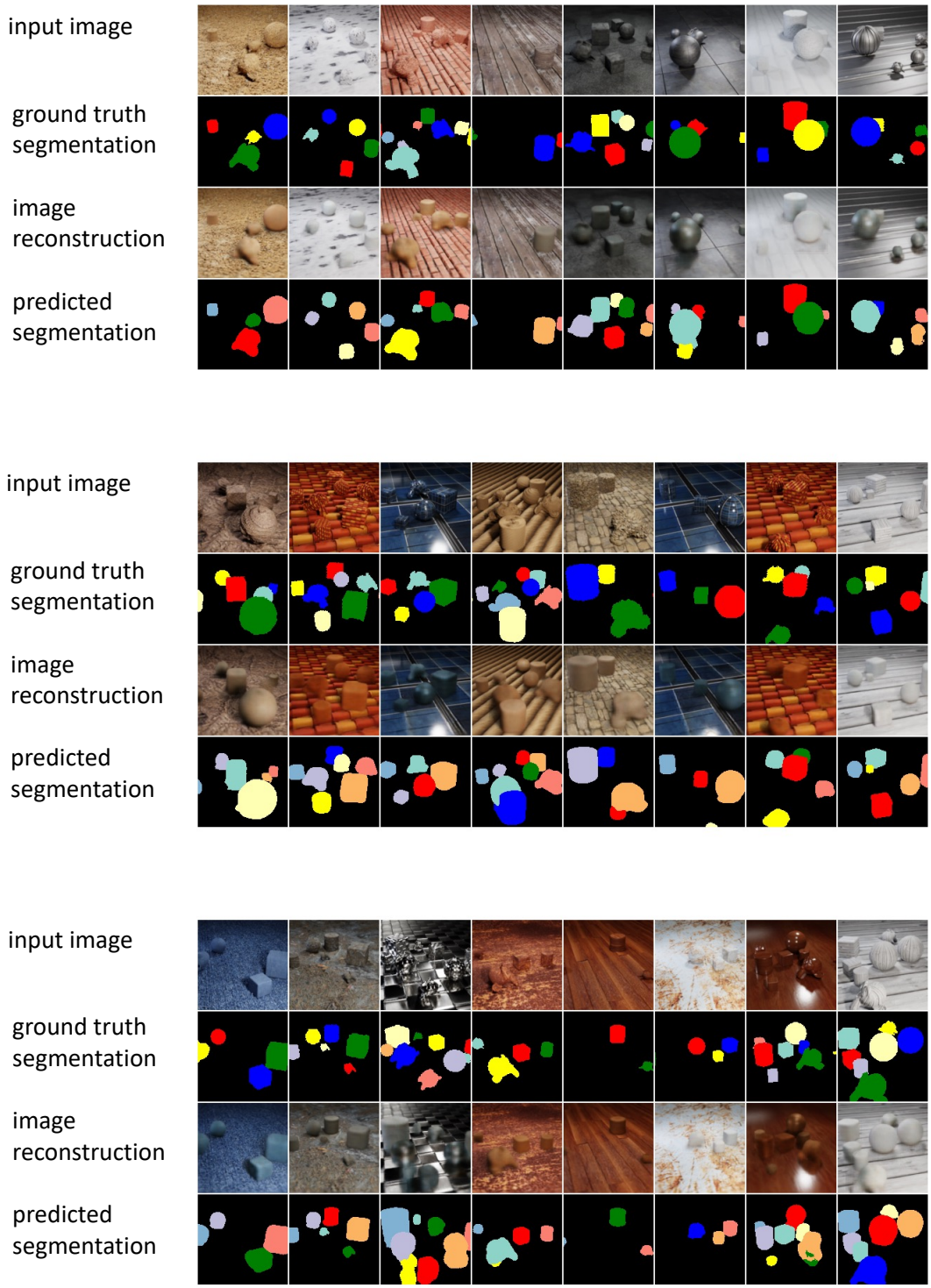


Figure 5.15: Examples of segmentation predictions on CAMO test dataset using a model trained on CLEVRTEX only

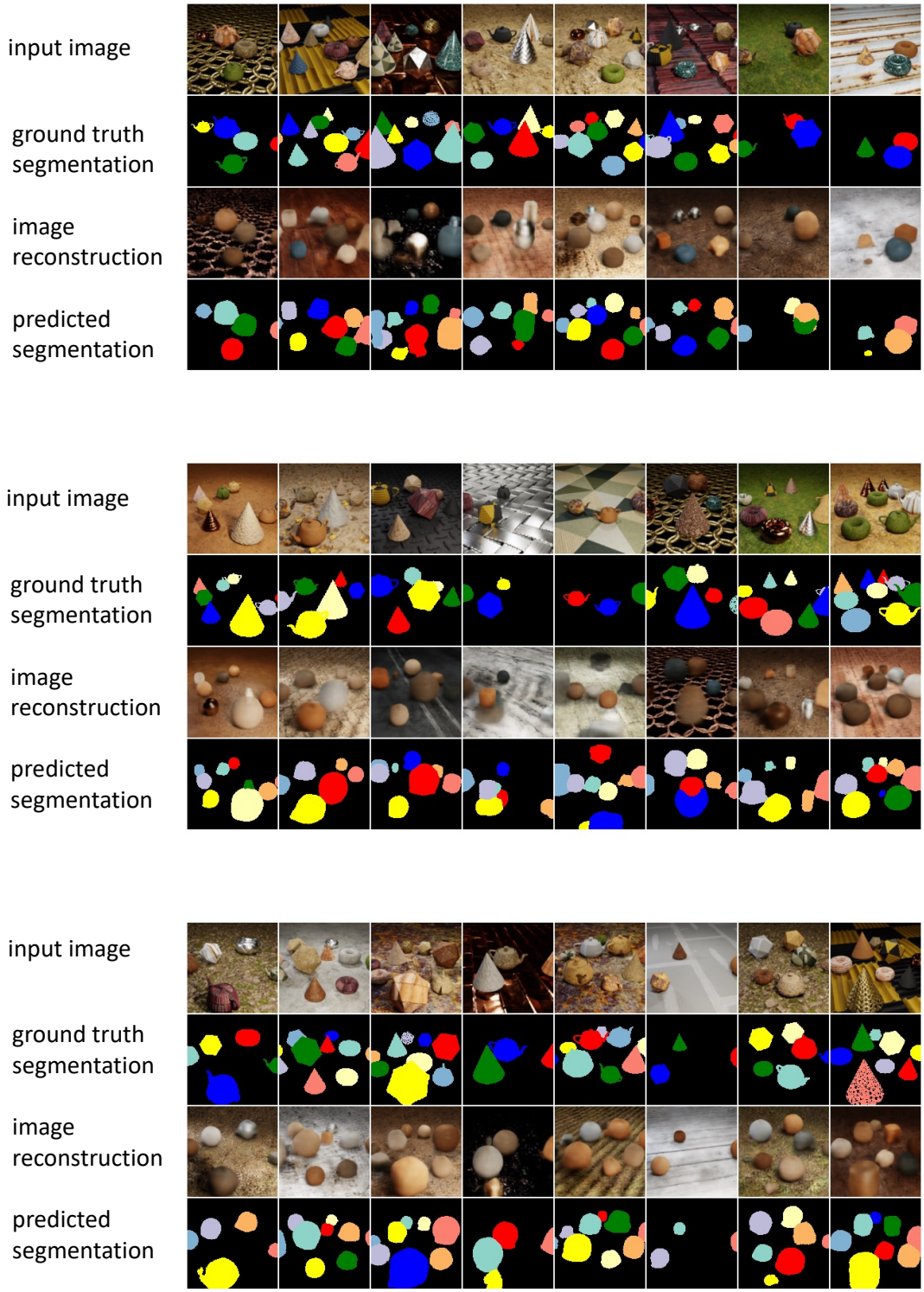


Figure 5.16: Examples of segmentation predictions on OOD test dataset using a model trained on CLEVRTEX only

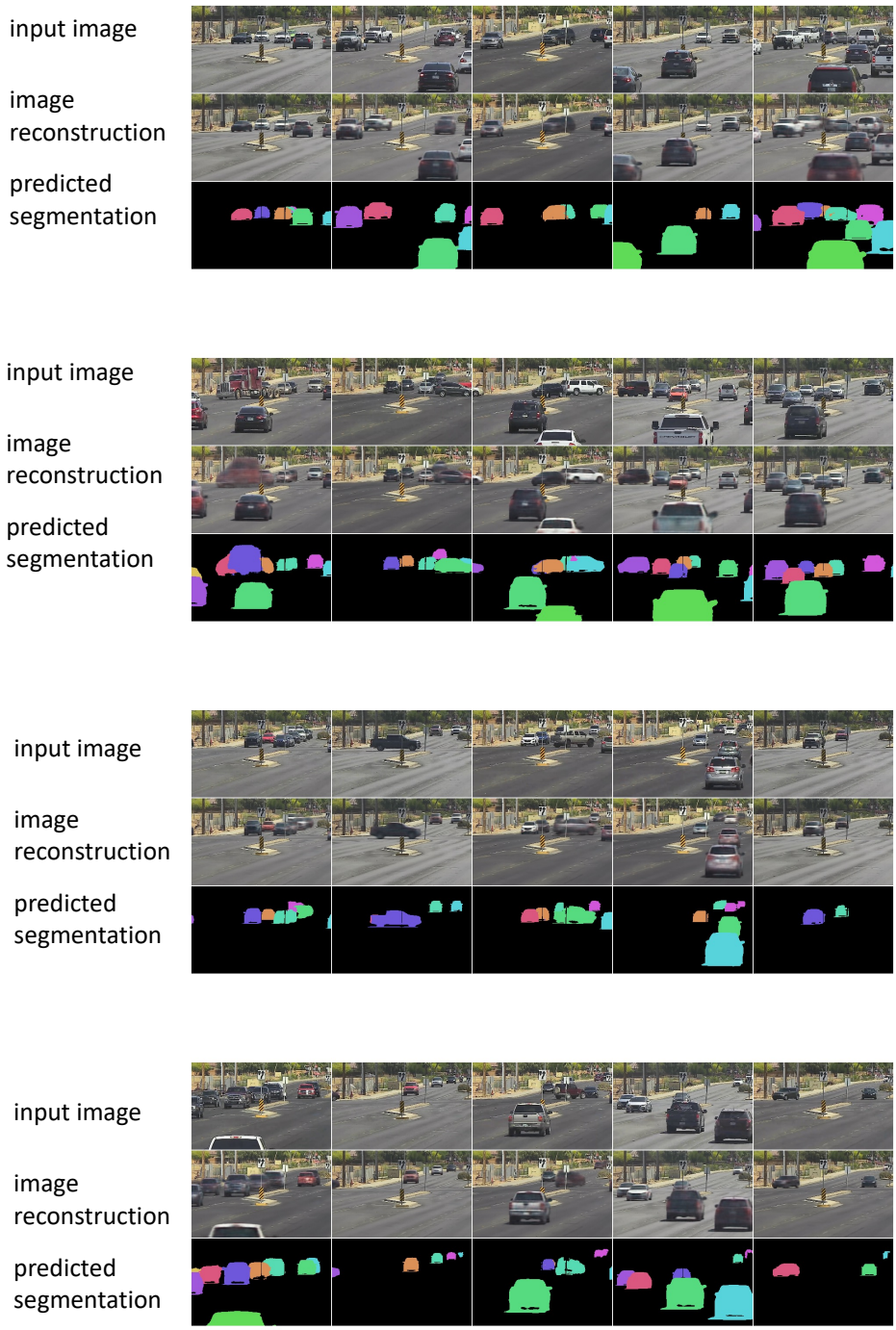


Figure 5.17: Examples of segmentation predictions on a real-world video extracted from a traffic webcam

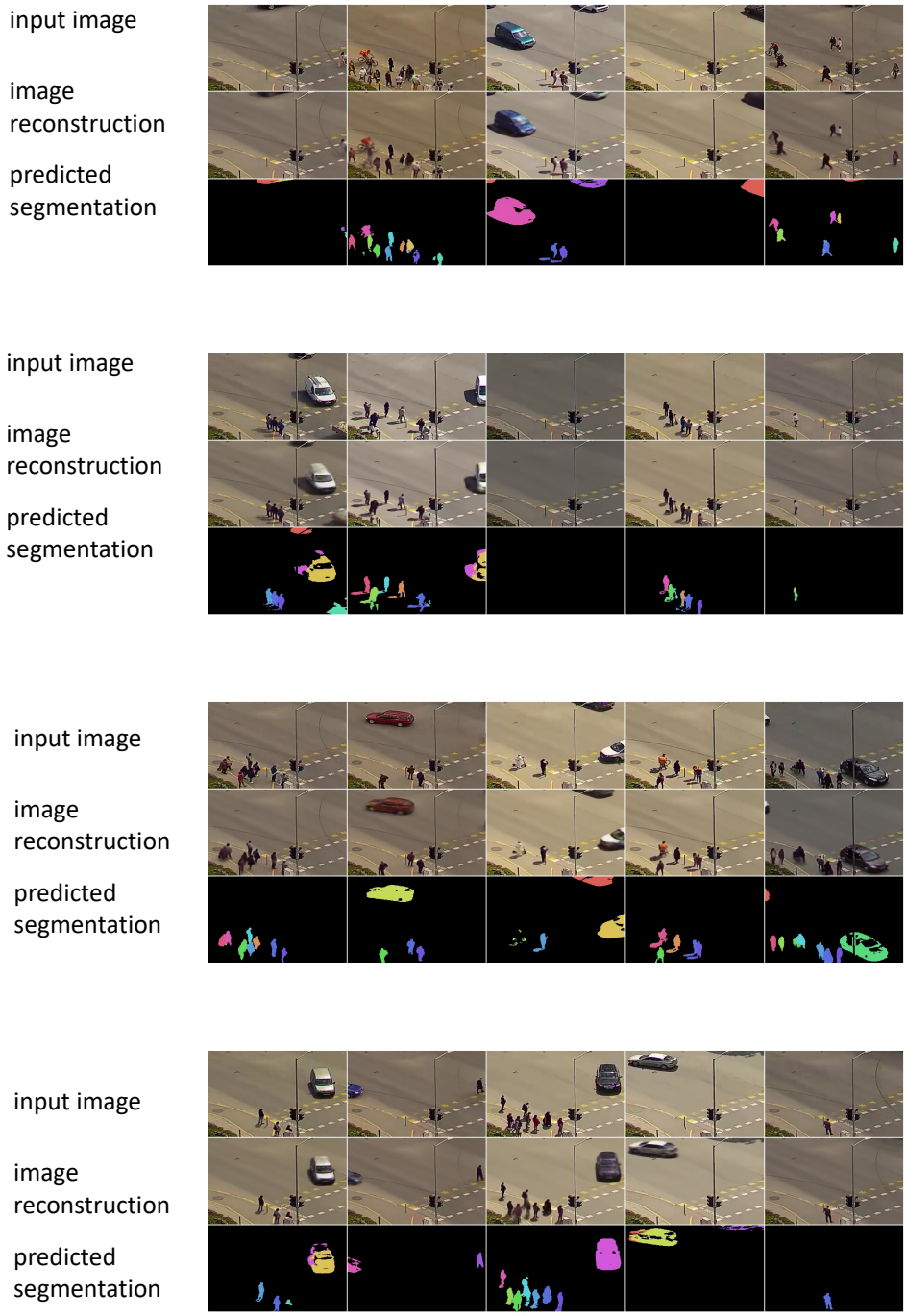


Figure 5.18: Examples of segmentation predictions on a real-world video extracted from a traffic webcam

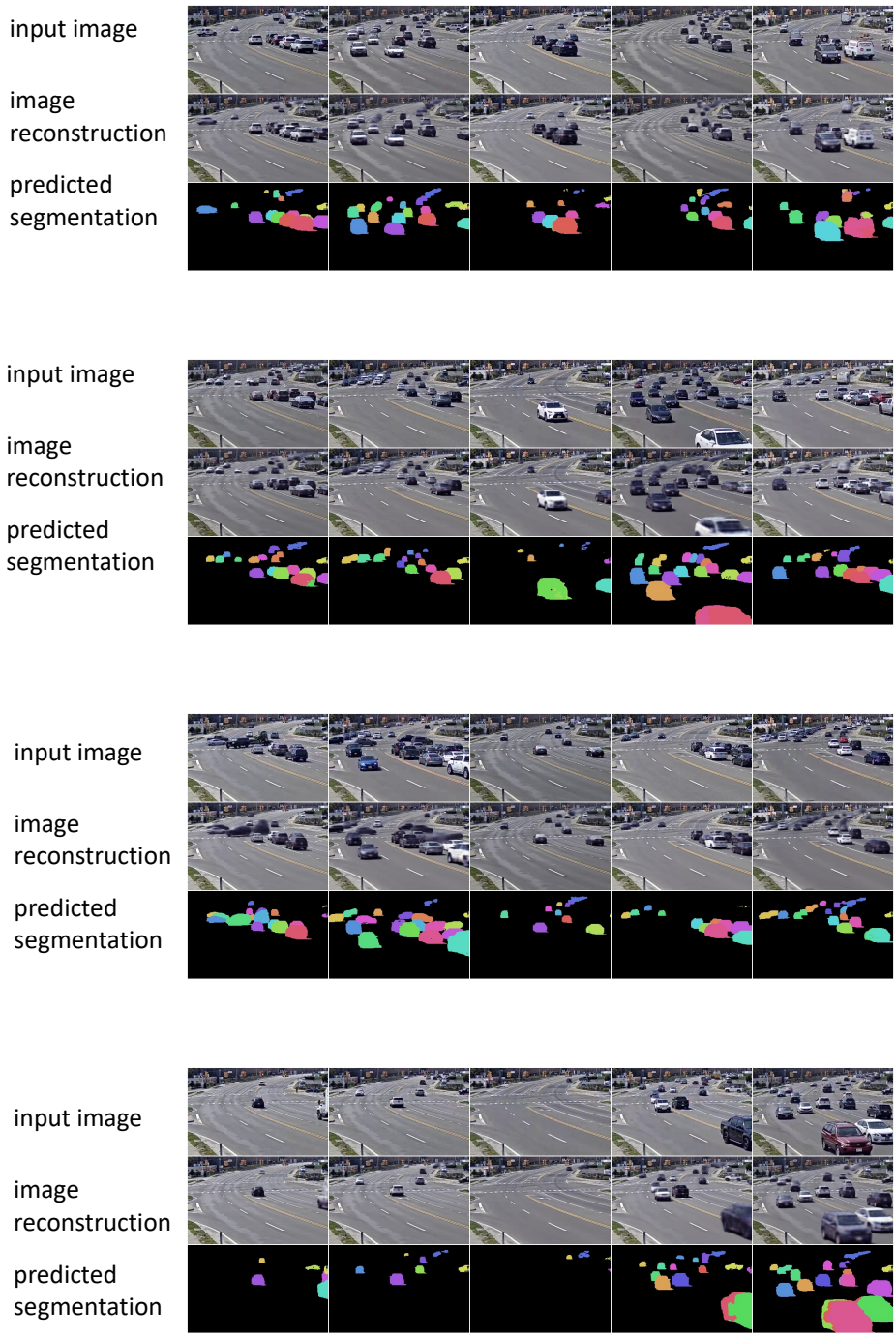


Figure 5.19: Examples of segmentation predictions on a real-world video extracted from a traffic webcam

Chapter 6

Discussion and conclusion

6.1 Résumé en français

Nous résumons dans cette conclusion les principales idées mises en oeuvre dans cette thèse : considérer la reconstruction d'arrière-plan comme une problème d'estimation robuste, modéliser l'arrière-plan comme une variété de petite dimension, utiliser un auto-encodeur pour non seulement reconstruire une image donnée en entrée, mais aussi pour prédire l'incertitude associée à cette reconstruction, et enfin prendre en compte les propriétés de symétrie et d'équivariance pour la conception de modèles d'apprentissage profond. Nous listons ensuite les principales limitations des modèles présentés ainsi que quelques pistes de recherches ultérieures.

6.2 Main ideas developed in this thesis

In this thesis, we have studied the tasks of fixed background reconstruction, dynamic background reconstruction, background/foreground segmentation, unsupervised multi-object segmentation and object-centric representation. The main ideas which have been developed to address these tasks can be summarized as follows:

- **Considering background reconstruction as a robust estimation problem**: Using a background estimate, it is possible to build a background/foreground segmentation. Inversely, one can benefit from a background/foreground segmentation to improve the accuracy of a background estimate by considering foreground pixels as outliers and excluding them from the estimation process. The associated iterative updates are the basis of the proposed fixed background reconstruction model.

- **Modeling the background as a low dimensional manifold**. It was already known that modeling the background as a low dimensional manifold is a powerful method to reconstruct dynamic backgrounds in video sequences. We have shown that this inductive bias is so efficient that it also allows to perform background reconstruction on videos taken from a moving camera and some non-video datasets.

- **Using an autoencoder to measure the accuracy loss associated to dimensionality reduction**. Deterministic autoencoders are usually trained to perform dimensionality reduction using a reconstruction loss. They are also able to directly provide an estimate of the level of accuracy loss caused by the dimensionality reduction, which can be useful for downstream applications.

- **The importance of symmetries and equivariance in the design of deep learning models**. Nearly all existing object detection models are not fully translation equivariant since they use a grid-based approach. The development of transformer-based models for vision applications which do not use any equivariance inductive bias could lead to the conclusion that these inductive biases are not really necessary. We have however designed an unsupervised model which is both simple and significantly more accurate than the state of the art by starting from the simple requirement that it should be translation equivariant and that the transformations applied to object feature vectors should be permutation equivariant.

6.3 Limitations and future works

- The management of shadows remains a significant issue for the proposed model.

- The creation of a benchmark on real-world scenes is a prerequisite to get a quantitative evaluation and optimization of the proposed model on real-world scenes.

- The proposed model is not a generative model, which may be an issue for applications such as future frames prediction.

- The proposed multi-object segmentation model does not use any temporal information such as optical flow. This design choice was justified during this thesis by the observation that available optical flow models were too slow or not able to properly handle small objects. Considering the progress of optical flow models, this choice may need to be reevaluated in the future.

- All the models which have been developed in this thesis are batch models: They assume that the full image dataset is available before starting the training phase, and no update is performed after the training is complete. Such a scheme may be acceptable for research purpose and offline image analysis, but not for real-world applications. We have seen that the background model alone is not robust to domain shift, but that the full model is robust to domain shift, so that an adaptation of the full model to real-time online simultaneous inference and training seems feasible, although it requires a significant computing power to handle high resolution images.

- Possible future works also include adapting this model to perform unsupervised object tracking on videos, which is currently an active field of study [Jiang et al., 2020, Crawford and Pineau, 2020, Kipf et al., 2022, Wu et al., 2021, Elsayed et al., 2022], as well as integrating it to reinforcement learning or VQA models, as discussed in section 2.6.

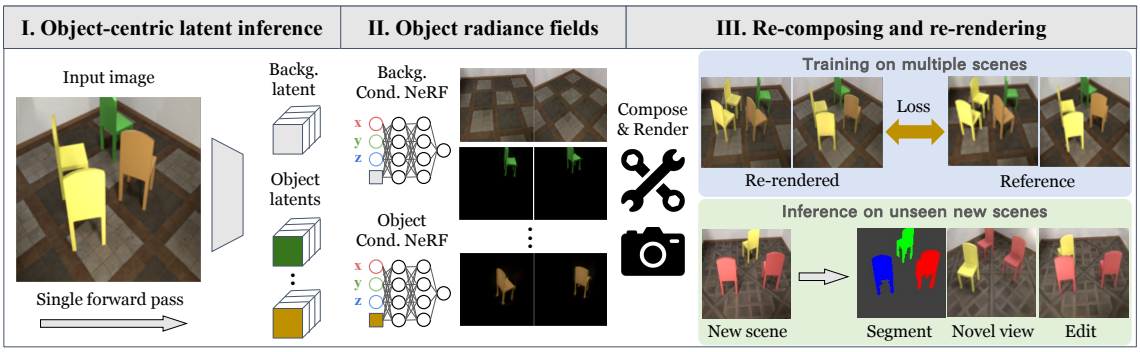


Figure 6.1: Overview of the uORF model, which uses conditional NeRFs as object and background generators. Source: [Yu et al., 2022]

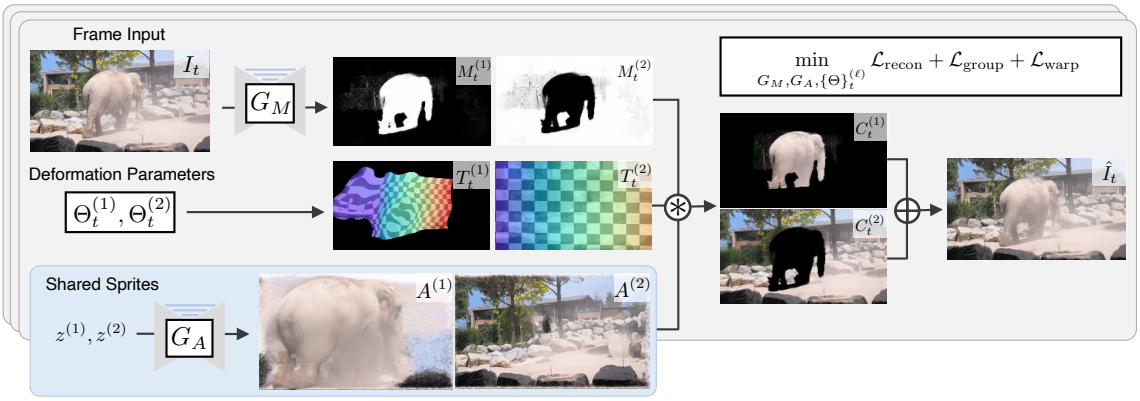


Figure 6.2: Overview of the model proposed in [Ye et al., 2022], which uses deformable sprites as object and background generators. Source: [Ye et al., 2022]

- The object generator of the proposed model is very simple, and the spatial transformer network considers only elementary 2D transformations. This may be insufficient to handle objects with complex shapes such as humans and animals or manage 3D object movements. Recently published models [Yu et al., 2022, Smith et al., 2022] show that it is possible to learn 3D object generators using conditional neural radiance fields (NeRF) in an unsupervised multi-object setting (Fig. 6.1). Another model [Ye et al., 2022] uses deformable sprites to generate objects with changing shapes such as animals (Fig. 6.2)

  The ideas developed in this thesis could also be applicable to unsupervised object discovery from point clouds data [Wang et al., 2022, You et al., 2022].

6.4 Publications

The results described in this report have led to the following publications:

- Bruno Sauvalle, Arnaud de La Fortelle. Fast and Accurate Background Reconstruction Using Background Bootstrapping, Journal of Imaging 8(1):9, January 2022
- Bruno Sauvalle, Arnaud de La Fortelle. Autoencoder-based background reconstruction and foreground segmentation with background noise estimation, IEEE/CVF Winter Conference on Applications of Computer Vision, (WACV) 2023, Waikoloa, HI, USA, January 2-7, 2023
- Bruno Sauvalle, Arnaud de La Fortelle. Unsupervised Multi-object Segmentation Using Attention and Soft-argmax, IEEE/CVF Winter Conference on Applications of Computer Vision, (WACV) 2023, Waikoloa, HI, USA, January 2-7, 2023

Chapter 7

Bibliography

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2281.

[Agarwala et al., 2004] Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. (2004). Interactive digital photomontage. *ACM SIGGRAPH 2004 Papers, SIGGRAPH 2004*, (June):294–302.

[Alec et al., 2019] Alec, R., Jeffrey, W., Rewon, C., David, L., Dario, A., and Ilya, S. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI Blog*.

[Ali Eslami et al., 2016] Ali Eslami, S. M., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. (2016). Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pages 3233–3241.

[Ali Eslami et al., 2018] Ali Eslami, S. M., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., Reichert, D. P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N., King, H., Hillier, C., Botvinick, M., Wierstra, D., Kavukcuoglu, K., and Hassabis, D. (2018). Neural scene representation and rendering. *Science*, 360(6394):1204–1210.

[Allebosch et al., 2016] Allebosch, G., Van Hamme, D., Deboeverie, F., Veelaert, P., and Philips, W. (2016). C-EFIC: Color and Edge Based Foreground Background Segmentation with Interior Classification. In Braz, J., Pettré, J., Richard, P., Kerren, A., Linsen, L., Battiato, S., and Imai, F., editors, *Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 433–454, Cham. Springer International Publishing.

[Ansuini et al., 2019] Ansuini, A., Laio, A., Macke, J. H., and Zoccolan, D. (2019). Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32.

[Asano et al., 2020] Asano, Y. M., Rupprecht, C., and Vedaldi, A. (2020). A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*.

[Assouel et al., 2022] Assouel, R., Castrejon, L., Courville, A., Ballas, N., and Bengio, Y. (2022). VIM: Variational Independent Modules for Video Prediction. In *Proceedings of Machine Learning Research Conference on Causal Learning and Reasoning*, volume 140, pages 1–19.

[Bae et al., 2021] Bae, G., Budvytis, I., and Cipolla, R. (2021). Estimating and Exploiting the Aleatoric Uncertainty in Surface Normal Estimation. *Proceedings of the IEEE International Conference on Computer Vision*, pages 13117–13126.

[Baltieri et al., 2010] Baltieri, D., Vezzani, R., and Cucchiara, R. (2010). Fast background initialization with recursive Hadamard transform. *Proceedings - IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010*, pages 165–171.

[Baradad et al., 2021] Baradad, M., Wulff, J., Wang, T., Isola, P., and Torralba, A. (2021). Learning to See by Looking at Noise. In *Advances in Neural Information Processing Systems*.

[Barnich and Van Droogenbroeck, 2009] Barnich, O. and Van Droogenbroeck, M. (2009). ViBE: A powerful random technique to estimate the background in video sequences. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 945–948.

[Barron, 2017] Barron, J. T. (2017). Continuously Differentiable Exponential Linear Units. *arXiv preprint*, 1704.07483.

[Behnaz et al., 2021] Behnaz, R., Amirreza, F., and Ostadabbas, S. (2021). DeepPBM: Deep Probabilistic Background Model Estimation from Video Sequences. In Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G. M., Mei, T., Bertini, M., Escalante, H. J., and Vezzani, R., editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 608–621, Cham. Springer International Publishing.

[Berjón et al., 2018] Berjón, D., Cuevas, C., Morán, F., and García, N. (2018). Real-time nonparametric background subtraction with tracking-based foreground update. *Pattern Recognition*, 74:156–170.

[Bertinetto et al., 2016] Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9914 LNCS:850–865.

[Bian et al., 2019] Bian, J. W., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M. M., and Reid, I. (2019). Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in Neural Information Processing Systems*, 32.

[Bianco et al., 2017] Bianco, S., Ciocca, G., and Schettini, R. (2017). How far can you get by combining change detection algorithms? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10484 LNCS:96–107.

[Bielski and Favaro, 2019] Bielski, A. and Favaro, P. (2019). Emergence of object segmentation in perturbed generative models. *Advances in Neural Information Processing Systems*.

[Bouwmans, 2014] Bouwmans, T. (2014). Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11-12:31–66.

[Bouwmans et al., 2019] Bouwmans, T., Javed, S., Sultana, M., and Jung, S. K. (2019). Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Networks*, 117:8–66.

[Bouwmans et al., 2017] Bouwmans, T., Maddalena, L., and Petrosino, A. (2017). Scene background initialization: A taxonomy. *Pattern Recognition Letters*, 96:3–11.

[Braham et al., 2018] Braham, M., Pierard, S., and Van Droogenbroeck, M. (2018). Semantic background subtraction. *Proceedings - International Conference on Image Processing, ICIP*, 2017-Septe:4552–4556.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[Bruna and Mallat, 2013] Bruna, J. and Mallat, S. (2013). Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886.

[Bunyak et al., 2007] Bunyak, F., Palaniappan, K., Nath, S. K., and Seetharaman, G. (2007). Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking. *Journal of Multimedia*, 2(4):20–33.

[Burgess et al., 2019] Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M. M., and Lerchner, A. (2019). MONet: Unsupervised Scene Decomposition and Representation. *CoRR*, abs/1901.1.

[Butler et al., 2005] Butler, D. E., Bove, V. M., and Sridharan, S. (2005). Real-Time Adaptive Foreground/Background Segmentation. *EURASIP Journal on Advances in Signal Processing*, (14).

[Candès et al., 2011] Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM*, 58(3).

[Carion et al., 2020] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12346 LNCS:213–229.

[Caron et al., 2021] Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging Properties in Self-Supervised Vision Transformers. *Proceedings of the IEEE International Conference on Computer Vision*, pages 9630–9640.

[Chandran et al., 2020] Chandran, P., Bradley, D., Gross, M., and Beeler, T. (2020). Attention-driven cropping for very high resolution facial landmark detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 5860–5869.

[Chang et al., 2004] Chang, R., Gandhi, T., and Trivedi, M. M. (2004). Vision modules for a multi-sensory bridge monitoring approach. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 971–976.

[Chen et al., 2019a] Chen, A. T. Y., Biglari-Abhari, M., and Wang, K. I. (2019a). SuperBE: computationally light background estimation with superpixels. *Journal of Real-Time Image Processing*, 16(6):2319–2335.

[Chen et al., 2019b] Chen, M., Artières, T., and Denoyer, L. (2019b). Unsupervised object segmentation by redrawing. *Advances in Neural Information Processing Systems*, 32.

[Chen et al., 2020a] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A Simple Framework for Contrastive Learning of Visual Representations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

[Chen et al., 2019c] Chen, W., Gao, J., Ling, H., Smith, E. J., Lehtinen, J., Jacobson, A., and Fidler, S. (2019c). Learning to predict 3D objects with an interpolation-based differentiable renderer. In *Advances in Neural Information Processing Systems*, volume 32.

[Chen et al., 2016] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2180–2188.

[Chen et al., 2020b] Chen, X., Fan, H., Girshick, R. B., and He, K. (2020b). Improved Baselines with Momentum Contrastive Learning. *ArXiv*, 2003.04297.

[Chen et al., 2021] Chen, X., Xie, S., and He, K. (2021). An Empirical Study of Training Self-Supervised Vision Transformers. *Proceedings of the IEEE International Conference on Computer Vision*, pages 9620–9629.

[Cheng et al., 2022] Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. (2022). Masked-attention Mask Transformer for Universal Image Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022-June:1280–1289.

[Cheng et al., 2021] Cheng, B., Schwing, A. G., and Kirillov, A. (2021). Per-Pixel Classification is Not All You Need for Semantic Segmentation. *Advances in Neural Information Processing Systems*, 22:17864–17875.

[Child, 2020] Child, R. (2020). Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images. In *International Conference on Learning Representations*.

[Clausse et al., 2019] Clausse, A., Benslimane, S., and De La Fortelle, A. (2019). Large-scale extraction of accurate vehicle trajectories for driving behavior learning. In *IEEE Intelligent Vehicles Symposium, Proceedings*, volume 2019-June, pages 2391–2396. Institute of Electrical and Electronics Engineers Inc.

[Cohen, 2005] Cohen, S. (2005). Background estimation as a labeling problem. *Proceedings of the IEEE International Conference on Computer Vision*, II:1034–1041.

[Colombari and Fusiello, 2010] Colombari, A. and Fusiello, A. (2010). Patch-based background initialization in heavily cluttered video. *IEEE Transactions on Image Processing*, 19(4):926–933.

[Colombari et al., 2005] Colombari, A., Informatica, D., Cristani, M., Informatica, D., Murino, V., Informatica, D., Fusiello, A., and Informatica, D. (2005). Exemplar-based Background Model Initialization Categories and Subject Descriptors. *Proc. VSSN 05, ACM New York, NY, USA*, (November).

[Crawford and Pineau, 2019] Crawford, E. and Pineau, J. (2019). Spatially Invariant Unsupervised Object Detection with Convolutional Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3412–3420.

[Crawford and Pineau, 2020] Crawford, E. and Pineau, J. (2020). Exploiting spatial invariance for scalable unsupervised object tracking. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 3684–3692.

[Creswell et al., 2021] Creswell, A., Kabra, R., Burgess, C., and Shanahan, M. (2021). Unsupervised Object-Based Transition Models for 3D Partially Observable Environments. *Advances in Neural Information Processing Systems*, 33:27344–27355.

[Cuevas et al., 2016] Cuevas, C., Yáñez, E. M., and García, N. (2016). Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. *Computer Vision and Image Understanding*, 152:103–117.

[De Gregorio and Giordano, 2015] De Gregorio, M. and Giordano, M. (2015). Background modeling by weightless neural networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9281:493–501.

[Detlefsen and Hauberg, 2019] Detlefsen, N. S. and Hauberg, S. (2019). Explicit disentanglement of appearance and perspective in generative models. *Advances in Neural Information Processing Systems*, 32.

[Devineau et al., 2018] Devineau, G., Polack, P., Altche, F., and Moutarde, F. (2018). Coupled Longitudinal and Lateral Control of a Vehicle using Deep Learning. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, volume 2018-Novem, pages 642–649.

[Devlin et al., 2019] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 4171–4186.

[Devon Hjelm et al., 2019] Devon Hjelm, R., Grewal, K., Bachman, P., Fedorov, A., Trischler, A., Lavoie-Marchildon, S., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019*.

[Diana and Bouwmans, 2010] Diana, F. and Bouwmans, T. (2010). Background modeling via a supervised subspace learning. In *International Conference on Image, Video Processing and Computer Vision*, pages pp.1–7. hal–00536017, Orlando.

[Ding et al., 2021] Ding, D., Hill, F., Santoro, A., Reynolds, M., and Botvinick, M. (2021). Attention over learned object embeddings enables complex visual reasoning. In *Advances in Neural Information Processing Systems*, volume 11, pages 9112–9124.

[Djerida et al., 2019] Djerida, A., Zhao, Z., and Zhao, J. (2019). Robust background generation based on an effective frames selection method and an efficient background estimation procedure (FSBE). *Signal Processing: Image Communication*, 78(February):21–31.

[Doersch et al., 2015] Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2015 Inter, pages 1422–1430.

[Donahue et al., 2014] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. In *31st International Conference on Machine Learning, ICML 2014*, volume 2, pages 988–996.

[Donahue and Simonyan, 2019] Donahue, J. and Simonyan, K. (2019). Large scale adversarial representation learning. *Advances in Neural Information Processing Systems*, 32.

[Dong et al., 2021] Dong, B., Zeng, F., Wang, T., Zhang, X., and Wei, Y. (2021). SOLQ: Segmenting Objects by Learning Queries. *Advances in Neural Information Processing Systems*, 26.

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[Duan et al., 2019] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). CenterNet: Keypoint triplets for object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:6568–6577.

[Edmonds and Karp, 1972] Edmonds, J. and Karp, R. M. (1972). Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journal of the ACM (JACM)*, 19(2):248–264.

[Elgammal et al., 2000] Elgammal, A., Harwood, D., and Davis, L. (2000). Non-parametric model for background subtraction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1843:751–767.

[Elguebaly and Bouguila, 2013] Elguebaly, T. and Bouguila, N. (2013). Finite asymmetric generalized Gaussian mixture models learning for infrared object detection. *Computer Vision and Image Understanding*, 117(12):1659–1671.

[Elsayed et al., 2022] Elsayed, G. F., Mahendran, A., van Steenkiste, S., Greff, K., Mozer, M. C., and Kipf, T. (2022). SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

[Emami et al., 2021] Emami, P., He, P., Ranka, S., and Rangarajan, A. (2021). Efficient Iterative Amortized Inference for Learning Symmetric and Disentangled Multi-Object Representations. *CoRR*, abs/2106.0.

[Engelcke et al., 2021] Engelcke, M., Jones, O. P., and Posner, I. (2021). GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement. In *Advances in Neural Information Processing Systems*, volume 10, pages 8085–8094.

[Engelcke et al., 2020] Engelcke, M., Kosiorek, A. R., Parker Jones, O., and Posner, I. (2020). GENESIS: Generative Scene Inference and Sampling with Object-centric Latent Representations. In *International Conference on Learning Representations*.

[Englesson and Azizpour, 2021] Englesson, E. and Azizpour, H. (2021). Consistency Regularization Can Improve Robustness to Label Noise. *Arxiv preprint arXiv:2110.01242*.

[Faro et al., 2011] Faro, A., Giordano, D., and Spampinato, C. (2011). Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1398–1412.

[Finn et al., 2016] Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. (2016). Deep spatial autoencoders for visuomotor learning. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June:512–519.

[Gao et al., 2021] Gao, T., Yao, X., and Chen, D. (2021). SimCSE: Simple Contrastive Learning of Sentence Embeddings. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 6894–6910.

[Garcia-Garcia et al., 2020] Garcia-Garcia, B., Bouwmans, T., and Silva, A. J. R. (2020). Background subtraction in real applications: Challenges, current models and future directions. *Computer Science Review*, 35:100204.

[Gatys et al., 2016] Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 2414–2423.

[Gidaris et al., 2018] Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

[Giraldo and Bouwmans, 2020] Giraldo, J. H. and Bouwmans, T. (2020). GraphBGS: Background subtraction via recovery of graph signals. *Proceedings - International Conference on Pattern Recognition*, pages 6881–6888.

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

[Goodfellow et al., 2020] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

[Goroshin et al., 2015] Goroshin, R., Mathieu, M., and Lecun, Y. (2015). Learning to linearize under uncertainty. *Advances in Neural Information Processing Systems*, pages 1234–1242.

[Goyal et al., 2021] Goyal, A., Lamb, A., Gampa, P., Beaudoin, P., Levine, S., Blundell, C., Bengio, Y., and Mozer, M. (2021). Factorizing Declarative and Procedural Knowledge in Structured, Dynamical Environments. *International Conference on Learning Representations*, pages 1–16.

[Greff et al., 2019] Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. (2019). Multi-object representation learning with iterative variational inference. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, pages 4317–4343.

[Greff et al., 2016] Greff, K., Rasmus, A., Berglund, M., Hao, T. H., Schmidhuber, J., and Valpola, H. (2016). Tagger: Deep unsupervised perceptual grouping. *Advances in Neural Information Processing Systems*, pages 4491–4499.

[Groth et al., 2018] Groth, O., Fuchs, F. B., Posner, I., and Vedaldi, A. (2018). ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11205 LNCS:724–739.

[Haines and Xiang, 2014] Haines, T. S. and Xiang, T. (2014). Background subtraction with dirichletprocess mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):670–683.

[Halfaoui et al., 2016] Halfaoui, I., Bouzaraa, F., and Urfalioglu, O. (2016). CNN-based initial background estimation. *Proceedings - International Conference on Pattern Recognition*, 0:101–106.

[He et al., 2022] He, K., Chen, X., Xie, S., Li, Y., Dollar, P., and Girshick, R. (2022). Masked Autoencoders Are Scalable Vision Learners. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022-June:15979–15988.

[He et al., 2020a] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020a). Momentum Contrast for Unsupervised Visual Representation Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9726–9735.

[He et al., 2020b] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2020b). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397.

[Henderson and Lampert, 2020] Henderson, P. and Lampert, C. H. (2020). Unsupervised object-centric video generation and decomposition in 3D. *Advances in Neural Information Processing Systems*.

[Hendrycks et al., 2019] Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, volume 32.

[Heravi et al., 2022] Heravi, N., Wahid, A., Lynch, C., Florence, P., Armstrong, T., Tompson, J., Sermanet, P., Bohg, J., and Dwibedi, D. (2022). Visuomotor Control in Multi-Object Scenes Using Object-Aware Representations. *CoRR*, abs/2205.0.

[Herzig et al., 2018] Herzig, R., Raboh, M., Chechik, G., Berant, J., and Globerson, A. (2018). Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Higgins et al., 2017] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). β-VAE: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–22.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*.

[Hofmann et al., 2012] Hofmann, M., Tiefenbacher, P., and Rigoll, G. (2012). Background segmentation with feedback: The pixel-based adaptive segmenter. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–43.

[Honari et al., 2018] Honari, S., Molchanov, P., Tyree, S., Vincent, P., Pal, C., and Kautz, J. (2018). Improving Landmark Localization with Semi-Supervised Learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1546–1555.

[Horan et al., 2021] Horan, D., Richardson, E., and Weiss, Y. (2021). When is Unsupervised Disentanglement Possible? *Advances in Neural Information Processing Systems*, 7:5150–5161.

[Hsiao and Leou, 2013] Hsiao, H.-H. and Leou, J.-J. (2013). Background initialization and foreground segmentation for bootstrapping video sequences. *EURASIP Journal on Image and Video Processing*, 2013(1):1.

[Hsieh et al., 2018] Hsieh, J. T., Liu, B., Huang, D. A., Fei-Fei, L., and Niebles, J. C. (2018). Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pages 517–526.

[Hu et al., 2020] Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2020). Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2011–2023.

[Huang and Belongie, 2017] Huang, X. and Belongie, S. (2017). Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 1510–1519.

[Huber, 1964] Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101.

[Hudson and Manning, 2019] Hudson, D. A. and Manning, C. D. (2019). Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32.

[Hwang et al., 2019] Hwang, J. J., Yu, S., Shi, J., Collins, M., Yang, T. J., Zhang, X., and Chen, L. C. (2019). SegSort: Segmentation by discriminative sorting of segments. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:7333–7343.

[Jaderberg et al., 2015] Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems*.

[Janner et al., 2019] Janner, M., Levine, S., Freeman, W. T., Tenenbaum, J. B., Finn, C., and Wu, J. (2019). Reasoning about physical interactions with object-oriented prediction and planning. *7th International Conference on Learning Representations, ICLR 2019*, pages 1–12.

[Javed et al., 2016] Javed, S., Jung, S. K., Mahmood, A., and Bouwmans, T. (2016). Motion-Aware Graph Regularized RPCA for background modeling of complex scenes. *Proceedings - International Conference on Pattern Recognition*, 0:120–125.

[Javed et al., 2019] Javed, S., Mahmood, A., Al-Maadeed, S., Bouwmans, T., and Jung, S. K. (2019). Moving Object Detection in Complex Scene Using Spatiotemporal Structured-Sparse RPCA. *IEEE Transactions on Image Processing*, 28(2):1007–1022.

[Javed et al., 2017] Javed, S., Mahmood, A., Bouwmans, T., and Jung, S. K. (2017). Background-Foreground Modeling Based on Spatiotemporal Sparse Subspace Clustering. *IEEE Transactions on Image Processing*, 26(12):5840–5854.

[Ji et al., 2019] Ji, X., Vedaldi, A., and Henriques, J. (2019). Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 9864–9873.

[Jiang and Ahn, 2020] Jiang, J. and Ahn, S. (2020). Generative neurosymbolic machines. In *Advances in Neural Information Processing Systems*.

[Jiang et al., 2020] Jiang, J., Janghorbani, S., de Melo, G., and Ahn, S. (2020). SCALOR: Generative World Models with Scalable Object Representations. In *International Conference on Learning Representations*.

[Jodoin et al., 2017] Jodoin, P. M., Maddalena, L., Petrosino, A., and Wang, Y. (2017). Extensive Benchmark and Survey of Modeling Methods for Scene Background Initialization. *IEEE Transactions on Image Processing*, 26(11):5244–5256.

[Johnson et al., 2017] Johnson, J., Fei-Fei, L., Hariharan, B., Zitnick, C. L., Van Der Maaten, L., and Girshick, R. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 1988–1997.

[Kabra et al., 2019] Kabra, R., Burgess, C., Matthey, L., Kaufman, R. L., Greff, K., Reynolds, M., and Lerchner, A. (2019). Multi-Object Datasets. https://github.com/deepmind/multi-object-datasets/.

[Kajo et al., 2020] Kajo, I., Kamel, N., and Ruichek, Y. (2020). Self-Motion-Assisted Tensor Completion Method for Background Initialization in Complex Video Sequences. *IEEE Transactions on Image Processing*, 29:1915–1928.

[Kajo et al., 2018] Kajo, I., Kamel, N., Ruichek, Y., and Malik, A. S. (2018). SVD-Based Tensor-Completion Technique for Background Initialization. *IEEE Transactions on Image Processing*, 27(6):3114–3126.

[Kalsotra and Arora, 2022] Kalsotra, R. and Arora, S. (2022). Background subtraction for moving object detection: explorations of recent developments and challenges. *The Visual Computer*, 38(12):4151–4178.

[Karazija et al., 2021] Karazija, L., Laina, I., and Rupprecht, C. (2021). ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation. (NeurIPS).

[Karras et al., 2019] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 4396–4405.

[Karras et al., 2020] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. pages 8107–8116. Institute of Electrical and Electronics Engineers (IEEE).

[Katircioglu et al., 2021] Katircioglu, I., Rhodin, H., Constantin, V., Sporri, J., Salzmann, M., and Fua, P. (2021). Self-supervised Human Detection and Segmentation via Background Inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12.

[Kendall and Gal, 2017] Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, pages 5575–5585.

[Khemakhem et al., 2020] Khemakhem, I., Kingma, D. P., Monti, R. P., and Hyvärinen, A. (2020). Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In Chiappa, S. and Calandra, R., editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2207–2217. PMLR.

[Kim et al., 2004] Kim, K., Chalidabhongse, T. H., Harwood, D., and Davis, L. (2004). Background modeling and subtraction by codebook construction. *Proceedings - International Conference on Image Processing, ICIP*, 2:3061–3064.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*.

[Kipf et al., 2022] Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. (2022). Conditional Object-Centric Learning from Video. In *International Conference on Learning Representations*.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

[Kolesnikov et al., 2019] Kolesnikov, A., Zhai, X., and Beyer, L. (2019). Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 1920–1929.

[Kosaraju et al., 2019] Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Hamid Rezatofighi, S., and Savarese, S. (2019). Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks. *Advances in Neural Information Processing Systems*, 32.

[Kosiorek et al., 2018] Kosiorek, A. R., Kim, H., Posner, I., and Teh, Y. W. (2018). Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pages 8606–8616.

[Kroeger et al., 2016] Kroeger, T., Timofte, R., Dai, D., and Van Gool, L. (2016). Fast optical flow using dense inverse search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9908 LNCS:471–488.

[Laugraud et al., 2016] Laugraud, B., Piérard, S., and Van Droogenbroeck, M. (2016). LaBGen-P: A pixel-level stationary background generation method based on LaBGen. *Proceedings - International Conference on Pattern Recognition*, 0:107–113.

[Laugraud et al., 2017] Laugraud, B., Piérard, S., and Van Droogenbroeck, M. (2017). LaBGen: A method based on motion detection for generating the background of a scene. *Pattern Recognition Letters*, 96:12–21.

[Laugraud et al., 2018] Laugraud, B., Piérard, S., and Van Droogenbroeck, M. (2018). Labgen-p-semantic: A first step for leveraging semantic segmentation in background generation. *Journal of Imaging*, 4(7):1–22.

[Laugraud and Van Droogenbroeck, 2017] Laugraud, B. and Van Droogenbroeck, M. (2017). Is a memoryless motion detection truly relevant for background generation with labgen? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10617 LNCS:443–454.

[Law and Deng, 2020] Law, H. and Deng, J. (2020). CornerNet: Detecting Objects as Paired Keypoints. *International Journal of Computer Vision*, 128(3):642–656.

[Lawson, 1961] Lawson, C. (1961). *Contribution to the Theory of Linear Least Maximum Approximations*. PhD thesis, University of California at Los Angeles.

[Lee et al., 2019] Lee, S.-h., Lee, G.-c., Yoo, J., and Kwon, S. (2019). WisenetMD: Motion Detection Using Dynamic Background Region Analysis. *Symmetry*, 11(5).

[Li et al., 2004] Li, H., Jiang, T., and Zhang, K. (2004). Efficient and robust feature extraction by maximum margin criterion. *Advances in Neural Information Processing Systems*, 17(1):157–165.

[Li et al., 2019] Li, X., Lai, T., Wang, S., Chen, Q., Yang, C., Chen, R., Lin, J., and Zheng, F. (2019). Weighted Feature Pyramid Networks for Object Detection. In *2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1500–1504.

[Lim and Keles, 2020] Lim, L. A. and Keles, H. Y. (2020). Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications*, 23(3):1369–1380.

[Lim and Yalim Keles, 2018] Lim, L. A. and Yalim Keles, H. (2018). Foreground segmentation using convolutional neural networks for multiscale feature encoding. *Pattern Recognition Letters*, 112:256–262.

[Lin et al., 2009] Lin, H. H., Liu, T. L., and Chuang, J. H. (2009). Learning a scene background model via classification. *IEEE Transactions on Signal Processing*, 57(5):1641–1654.

[Lin et al., 2020] Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. (2020). SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. In *International Conference on Learning Representations*.

[Liu et al., 2016] Liu, W., Cai, Y., Zhang, M., Li, H., and Gu, H. (2016). Scene background estimation based on temporal median filter with

Gaussian filtering. *Proceedings - International Conference on Pattern Recognition*, 0:132–136.

[Liu et al., 2015] Liu, X., Zhao, G., Yao, J., and Qi, C. (2015). Background subtraction based on low-rank and structured sparse decomposition. *IEEE Transactions on Image Processing*, 24(8):2502–2514.

[Locatello et al., 2019] Locatello, F., Bauer, S., Lucie, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *36th International Conference on Machine Learning, ICML 2019*, pages 7247–7283.

[Locatello et al., 2020] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. (2020). Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, (NeurIPS).

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

[Lu et al., 2020] Lu, X., Wang, W., Shen, J., Tai, Y., Crandall, D. J., and Hoi, S. H. (2020). Learning Video Object Segmentation From Unlabeled Videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8957–8967, Los Alamitos, CA, USA. IEEE Computer Society.

[Luo et al., 2016] Luo, G., Zhu, Y., Li, Z., and Zhang, L. (2016). A Hole Filling Approach Based on Background Reconstruction for View Synthesis in 3D Video. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:1781–1789.

[Luvizon et al., 2019] Luvizon, D. C., Tabia, H., and Picard, D. (2019). Human pose regression by combining indirect part detection and contextual information. *Computers and Graphics (Pergamon)*, 85:15–22.

[Maddalena and Petrosino, 2008a] Maddalena, L. and Petrosino, A. (2008a). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177.

[Maddalena and Petrosino, 2008b] Maddalena, L. and Petrosino, A. (2008b). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177.

[Maddalena and Petrosino, 2012] Maddalena, L. and Petrosino, A. (2012). The SOBS algorithm: What are the limits? *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 21–26.

[Maddalena and Petrosino, 2015] Maddalena, L. and Petrosino, A. (2015). Towards benchmarking scene background initialization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9281:469–476.

[Maddalena and Petrosino, 2016] Maddalena, L. and Petrosino, A. (2016). Extracting a background image by a multi-modal scene background model. *Proceedings - International Conference on Pattern Recognition*, 0:143–148.

[Mairal et al., 2010] Mairal, J., Jenatton, R., Obozinski, G., and Bach, F. (2010). Network flow algorithms for structured sparsity. *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, pages 1–9.

[Mandal and Vipparthi, 2020] Mandal, M. and Vipparthi, S. K. (2020). Scene Independency Matters: An Empirical Study of Scene Dependent and Scene Independent Evaluation for CNN-Based Change Detection. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–14.

[Mandal and Vipparthi, 2022] Mandal, M. and Vipparthi, S. K. (2022). An Empirical Review of Deep Learning Frameworks for Change Detection: Model Design, Experimental Frameworks, Challenges and Research Needs. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6101–6122.

[Maron et al., 2020] Maron, H., Litany, O., Chechik, G., and Fetaya, E. (2020). On Learning Sets of Symmetric Elements. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

[Mikolov et al., 2013a] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

[Mikolov et al., 2013b] Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

[Mildenhall et al., 2020] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12346 LNCS:405–421.

[Min et al., 2021] Min, C. H., Bae, J., Lee, J., and Kim, Y. M. (2021). GATSBI: Generative Agent-centric Spatio-temporal Object Interaction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3073–3082.

[Misra and Van Der Maaten, 2020] Misra, I. and Van Der Maaten, L. (2020). Self-Supervised Learning of Pretext-Invariant Representations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Mondéjar-Guerra et al., 2020] Mondéjar-Guerra, V., Rouco, J., Novo, J., and Ortega, M. (2020). An end-to-end deep learning approach for simultaneous background modeling and subtraction. *30th British Machine Vision Conference 2019, BMVC 2019*, pages 1–12.

[Monnier et al., 2021] Monnier, T., Vincent, E., Ponce, J., and Aubry, M. (2021). Unsupervised layered image decomposition into object prototypes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8640–8650.

[Moreau et al., 2022] Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., and De La Fortelle, A. (2022). CoordiNet: Uncertainty-aware pose regressor for reliable vehicle localization. *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*, pages 1848–1857.

[Mseddi et al., 2019] Mseddi, W. S., Jmal, M., and Attia, R. (2019). Real-time scene background initialization based on spatio-temporal neighborhood exploration. *Multimedia Tools and Applications*, 78(6):7289–7319.

[Mukherjee and Wu, 2012] Mukherjee, D. and Wu, Q. M. (2012). Real-time video segmentation using student's t mixture model. *Procedia Computer Science*, 10:153–160.

[Newell et al., 2016] Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9912 LNCS:483–499.

[Nguyen et al., 2022] Nguyen, D. Q., Nguyen, T. D., and Phung, D. (2022). Universal Self-Attention Network for Graph Classification. In *WWW '22*, New York, NY, USA. Association for Computing Machinery.

[Nix and Weigend, 1994] Nix, D. A. and Weigend, A. S. (1994). Estimating the mean and variance of the target probability distribution. *IEEE International Conference on Neural Networks - Conference Proceedings*, 1:55–60.

[N.M. et al., 2000] N.M., O., B., R., and A.P., P. (2000). A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843.

[Ortego et al., 2016] Ortego, D., SanMiguel, J. C., and Martínez, J. M. (2016). Rejection based multipath reconstruction for background estimation in video sequences with stationary objects. *Computer Vision and Image Understanding*, 147:23–37.

[Oyallon et al., 2018] Oyallon, E., Zagoruyko, S., Huang, G., Komodakis, N., Lacoste-Julien, S., Blaschko, M. B., and Belilovsky, E. (2018). Scattering Networks for Hybrid Representation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 11.

[Ozair et al., 2019] Ozair, S., Lynch, C., Bengio, Y., van den Oord, A., Levine, S., and Sermanet, P. (2019). Wasserstein dependency measure for representation learning. *Advances in Neural Information Processing Systems*, 32.

[Pardàs and Canet, 2021] Pardàs, M. and Canet, G. (2021). Refinement network for unsupervised on the scene foreground segmentation. *European Signal Processing Conference*, Jan:705–709.

[Pathak et al., 2016] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context Encoders: Feature Learning by Inpainting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 2536–2544.

[Pham and Smeulders, 2006] Pham, T. V. and Smeulders, A. W. M. (2006). Efficient projection pursuit density estimation for background subtraction. In *IEEE Intern Workshop on Visual Surveillance*.

[Piccardi, 2004] Piccardi, M. (2004). Background subtraction techniques: A review. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 4:3099–3104.

[Preechakul et al., 2022] Preechakul, K., Chatthee, N., Wizadwongsa, S., and Suwajanakorn, S. (2022). Diffusion Autoencoders: Toward a Meaningful and Decodable Representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2022-June, pages 10609–10619.

[Prost et al., 2022] Prost, J., Houdard, A., Papadakis, N., and Almansa, A. (2022). Diverse super-resolution with pretrained deep hierarchical VAEs. *arXiv preprint*, (2205.10347).

[Qi et al., 2017] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Jan, pages 77–85.

[Raboh et al., 2020] Raboh, M., Herzig, R., Berant, J., Chechik, G., and Globerson, A. (2020). Differentiable scene graphs. In *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pages 1477–1486.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

[Radford et al., 2016] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

[Rahmon et al., 2020] Rahmon, G., Bunyak, F., Seetharaman, G., and Palaniappan, K. (2020). Motion U-Net: Multi-cue encoder-decoder network for motion segmentation. *Proceedings - International Conference on Pattern Recognition*, pages 8125–8132.

[Ramachandran et al., 2019] Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. (2019). *Stand-Alone Self-Attention in Vision Models*. Curran Associates Inc., Red Hook, NY, USA.

[Ramesh et al., 2022] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical Text-Conditional Image Generation with CLIP Latents. Technical Report Arxiv preprint:2204.06125.

[Razavian et al., 2014] Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 512–519.

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

[Ren et al., 2018] Ren, X., Wang, D., Laskey, M., and Goldberg, K. (2018). Learning Traffic Behaviors by Extracting Vehicle Trajectories from Online Video Streams. *IEEE International Conference on Automation Science and Engineering*, 2018:1276–1283.

[Rezaei et al., 2020] Rezaei, B., Farnoosh, A., and Ostadabbas, S. (2020). G-LBM: Generative Low-Dimensional Background Model Estimation from Video Sequences. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12357 LNCS:293–310.

[Ridder et al., 1995] Ridder, C., Munkelt, O., and Kirchner, H. (1995). Adaptive Background Estimation and Foreground Detection using Kalman Filtering. *Proceedings of the International Conference on Recent Advances in Mechatronics (ICRAM 1995)*, pages 193–199.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

[Rosenfeld and Pfaltz, 1966] Rosenfeld, A. and Pfaltz, J. L. (1966). Sequential Operations in Digital Picture Processing. *Journal of the ACM (JACM)*, 13(4):471–494.

[Roynard et al., 2018a] Roynard, X., Deschaud, J. E., and Goulette, F. (2018a). Paris-lille-3D: A point cloud dataset for urban scene segmentation and classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June:2108–2111.

[Roynard et al., 2018b] Roynard, X., Deschaud, J.-e., Goulette, F., Roynard, X., Deschaud, J.-e., Goulette, F., Cloud, P., Roynard, X., Deschaud, J.-e., and Goulette, F. (2018b). Classification of Point Cloud for Road Scene Understanding with Multiscale Voxel Deep Network. In *10th workshop on Planning, Perceptionand Navigation for Intelligent Vehicules PPNIV'2018*.

[Sanderson et al., 2011] Sanderson, C., Reddy, V., and Lovell, B. C. (2011). A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts. *Eurasip Journal on Image and Video Processing*, 2011.

[Saseendran et al., 2021] Saseendran, A., Skubch, K., Falkner, S., and Keuper, M. (2021). Shape your Space: A Gaussian Mixture Regularization Approach to Deterministic Autoencoders. *Advances in Neural Information Processing Systems*, 9:7319–7332.

[Sauer et al., 2022] Sauer, A., Schwarz, K., and Geiger, A. (2022). StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings), August, 2022, Vancouver, BC, Canada*, 1(1):1–10.

[Seitzer et al., 2022] Seitzer, M., Tavakoli, A., Antic, D., and Martius, G. (2022). On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In *International Conference on Learning Representations*.

[Shahbaz et al., 2015] Shahbaz, A., Hariyono, J., and Jo, K. H. (2015). Evaluation of background subtraction algorithms for video surveillance. *2015 Frontiers of Computer Vision, FCV 2015*.

[Sitzmann et al., 2019] Sitzmann, V., Zollhöfer, M., and Wetzstein, G. (2019). Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32.

[Smirnov et al., 2021] Smirnov, D., Gharbi, M., Fisher, M., Guizilini, V., Efros, A. A., and Solomon, J. (2021). MarioNette: Self-Supervised Sprite Learning. *Advances in Neural Information Processing Systems*, 7(NeurIPS):5494–5505.

[Smith et al., 2022] Smith, C., Yu, H.-X., Zakharov, S., Durand, F., Tenenbaum, J. B., Wu, J., and Sitzmann, V. (2022). Unsupervised Discovery and Composition of Object Light Fields. *arXiv preprint*, (arXiv :2205.03923).

[Sobral, 2013] Sobral, A. (2013). BGSLibrary: An OpenCV C++ Background Subtraction Library. *IX Workshop de Visao Computacional (WVC'2013)*, (JUNE 2013):1–3.

[Sobral et al., 2015] Sobral, A., Bouwmans, T., and Zahzah, E. H. (2015). Comparison of matrix completion algorithms for background initialization in videos. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9281:510–518.

[Sobral and hadi Zahzah, 2017] Sobral, A. and hadi Zahzah, E. (2017). Matrix and tensor completion algorithms for background model initialization: A comparative evaluation. *Pattern Recognition Letters*, 96:22–33.

[Sohl-Dickstein et al., 2015] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.

[Song et al., 2021] Song, J., Meng, C., and Ermon, S. (2021). Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

[Sorrenson et al., 2020] Sorrenson, P., Rother, C., and Köthe, U. (2020). Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN). In *International Conference on Learning Representations*.

[St-Charles et al., 2015] St-Charles, P. L., Bilodeau, G. A., and Bergevin, R. (2015). SuBSENSE: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373.

[St-Charles et al., 2016] St-Charles, P. L., Bilodeau, G. A., and Bergevin, R. (2016). Universal Background Subtraction Using Word Consensus Models. *IEEE Transactions on Image Processing*, 25(10):4768–4781.

[Stauffer and Grimson, 1999] Stauffer, C. and Grimson, W. E. (1999). Adaptive background mixture models for real-time tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:246–252.

[Stelzner et al., 2019] Stelzner, K., Peharz, R., and Kersting, K. (2019). Faster Attend-Infer-Repeat with Tractable Probabilistic Models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 5966–5975.

[Sultana et al., 2019] Sultana, M., Mahmood, A., Javed, S., and Jung, S. K. (2019). Unsupervised deep context prediction for background estimation and foreground segmentation. *Machine Vision and Applications*, 30(3):375–395.

[Sun et al., 2018] Sun, X., Xiao, B., Wei, F., Liang, S., and Wei, Y. (2018). Integral human pose regression. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11210 LNCS:536–553.

[Tang et al., 2022] Tang, Q., Xiangyu, Z., Zhen, L., and Zhang, Z. (2022). Object Dynamics Distillation for Scene Decomposition and Representation. In *International Conference on Learning Representations*.

[Tao et al., 2017] Tao, Y., Palasek, P., Ling, Z., and Patras, I. (2017). Background modelling based on generative unet. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*, (August).

[Teng and Wang, 2022] Teng, Y. and Wang, L. (2022). Structured Sparse R-CNN for Direct Scene Graph Generation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19415–19424, Los Alamitos, CA, USA. IEEE Computer Society.

[Tezcan et al., 2021] Tezcan, M. O., Ishwar, P., and Konrad, J. (2021). BSUV-Net 2.0: Spatio-Temporal Data Augmentations for Video-Agnostic Supervised Background Subtraction. *IEEE Access*, 9:53849–53860.

[Thomas et al., 2019] Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., and Guibas, L. (2019). KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 6410–6419.

[Tiulpin et al., 2019] Tiulpin, A., Melekhov, I., and Saarakkala, S. (2019). KNEEL: Knee anatomical landmark localization using hourglass networks. In *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pages 352–361.

[Touvron et al., 2020] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2020). Training data-efficient image transformers distillation through attention. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR.

[Touvron et al., 2022] Touvron, H., Cord, M., and Jégou, H. (2022). DeiT III: Revenge of the ViT. In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T., editors, *ECCV 2022*, volume 13684 LNCS, pages 516–533, Cham. Springer Nature Switzerland.

[Touvron et al., 2021] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jegou, H. (2021). Going deeper with Image Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 32–42, Los Alamitos, CA, USA. IEEE Computer Society.

[Toyama et al., 1999] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. *Proceedings of the IEEE International Conference on Computer Vision*, 1:255–261.

[Vacavant et al., 2013] Vacavant, A., Chateau, T., Wilhelm, A., and Lequièvre, L. (2013). A benchmark dataset for outdoor foreground/background extraction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7728 LNCS(PART 1):291–300.

[van den Oord et al., 2018] van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation Learning with Contrastive Predictive Coding. *CoRR*, abs/1807.0.

[Van Steenkiste et al., 2018] Van Steenkiste, S., Greff, K., Chang, M., and Schmidhuber, J. (2018). Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pages 1–15.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009.

[Veerapaneni et al., 2020] Veerapaneni, R., Co-Reyes, J. D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J., and Levine, S. (2020).

Entity Abstraction in Visual Model-Based Reinforcement Learning. In Kaelbling, L. P., Kragic, D., and Sugiura, K., editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1439–1456. PMLR.

[Veličković et al., 2018] Veličković, P., Casanova, A., Liò, P., Cucurull, G., Romero, A., and Bengio, Y. (2018). Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

[Vinyals et al., 2016] Vinyals, O., Bengio, S., and Kudlur, M. (2016). Order matters: Sequence to sequence for sets. In *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

[Wang et al., 2019a] Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., and Li, H. (2019a). Unsupervised deep tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 1308–1317.

[Wang et al., 2014a] Wang, R., Bunyak, F., Seetharaman, G., and Palaniappan, K. (2014a). Static and moving object detection using flux tensor with split gaussian models - Wang et al. - 2014 - IEEE Computer Society Conference.pdf. *IEEE Change Detection Workshop, CVPR*, pages 414–418.

[Wang et al., 2022] Wang, T., Liu, M., and Ng, K. S. (2022). Spatially Invariant Unsupervised 3D Object-Centric Learning And Scene Decomposition. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 120–135, Berlin, Heidelberg. Springer-Verlag.

[Wang et al., 2019b] Wang, X., He, J., and Ma, L. (2019b). Exploiting Local and Global Structure for Point Cloud Semantic Segmentation with Contextual Point Representations. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Wang et al., 2014b] Wang, Y., Jodoin, P. M., Porikli, F., Konrad, J., Benezeth, Y., and Ishwar, P. (2014b). CDnet 2014: An expanded change detection benchmark dataset. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 393–400.

[Wren et al., 1997] Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: Real-Time Tracking of the Human Body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785.

[Wright et al., 2009] Wright, J., Peng, Y., Ma, Y., Ganesh, A., and Rao, S. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pages 2080–2088.

[Wu and He, 2020] Wu, Y. and He, K. (2020). Group Normalization. *International Journal of Computer Vision*, 128(3):742–755.

[Wu et al., 2021] Wu, Y., Jones, O. P., Engelcke, M., and Posner, I. (2021). APEX: Unsupervised, Object-Centric Scene Segmentation and Tracking for Robot Manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3375–3382.

[Xie et al., 2021] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021). SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *Advances in Neural Information Processing Systems*, volume 15, pages 12077–12090.

[Xin et al., 2015] Xin, B., Tian, Y., Wang, Y., and Gao, W. (2015). Background Subtraction via generalized fused lasso foreground modeling. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:4676–4684.

[Xu et al., 2017] Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. (2017). Scene graph generation by iterative message passing. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 3097–3106.

[Xu et al., 2019a] Xu, K., Jegelka, S., Hu, W., and Leskovec, J. (2019a). How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019*.

[Xu and Huang, 2008] Xu, X. and Huang, T. S. (2008). A loopy belief propagation approach for robust background estimation. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

[Xu et al., 2020] Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. (2020). A Theory of Usable Information under Computational Constraints. In *International Conference on Learning Representations*.

[Xu et al., 2019b] Xu, Z., Min, B., and Cheung, R. C. (2019b). A robust background initialization algorithm with superpixel motion detection. *Signal Processing: Image Communication*, 71:1–12.

[Yang et al., 2017] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., and Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:4076–4084.

[Yang et al., 2019] Yang, Y., Loquercio, A., Scaramuzza, D., and Soatto, S. (2019). Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 879–888.

[Ye et al., 2022] Ye, V., Li, Z., Tucker, R., Kanazawa, A., and Snavely, N. (2022). Deformable Sprites for Unsupervised Video Decomposition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2647–2656, Los Alamitos, CA, USA. IEEE Computer Society.

[Yosinski et al., 2014] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 4(January):3320–3328.

[You et al., 2022] You, Y., Luo, K., Phoo, C., Chao, W., Sun, W., Hariharan, B., Campbell, M., and Weinberger, K. Q. (2022). Learning to Detect Mobile Objects from LiDAR Scans Without Labels. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1120–1130, Los Alamitos, CA, USA. IEEE Computer Society.

[Yu and Koltun, 2016] Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

[Yu et al., 2022] Yu, H.-X., Guibas, L. J., and Wu, J. (2022). Unsupervised Discovery of Object Radiance Fields. In *International Conference on Learning Representations*.

[Yu and Guo, 2019] Yu, L. and Guo, W. (2019). A Robust Background Initialization Method Based on Stable Image Patches. *Proceedings 2018 Chinese Automation Congress, CAC 2018*, pages 980–984.

[Yuan et al., 2020] Yuan, Y., Chen, X., and Wang, J. (2020). Object-Contextual Representations for Semantic Segmentation. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 173–190, Cham. Springer International Publishing.

[Zadaianchuk et al., 2021] Zadaianchuk, A., Seitzer, M., and Martius, G. (2021). Self-supervised Visual Reinforcement Learning with Object-centric Representations. In *International Conference on Learning Representations*.

[Zaheer et al., 2017] Zaheer, M., Kottur, S., Ravanbhakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. J. (2017). Deep sets. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 3392–3402.

[Zeng et al., 2019] Zeng, D., Chen, X., Zhu, M., Goesele, M., and Kuijper, A. (2019). Background Subtraction with Real-Time Semantic Segmentation. *IEEE Access*, 7:153869–153884.

[Zhang et al., 2016] Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful Image Colorization. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 649–666, Cham. Springer International Publishing.

[Zhang et al., 2009] Zhang, S., Yao, H., Liu, S., Zhang, S., Yao, H., Liu, S., Background, D., Based, S., Zhang, S., Yao, H., and Liu, S. (2009). Dynamic Background Subtraction Based on Local Dependency Histogram. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(7):1397–1419.

[Zhang et al., 2019] Zhang, Y., Hare, J., and Prügel-Bennett, A. (2019). *Deep set prediction networks*, volume 32.

[Zhao et al., 2022] Zhao, X., Wang, G., He, Z., and Jiang, H. (2022). A survey of moving object detection methods: A practical perspective. *Neurocomputing*, 503:28–48.

[Zheng et al., 2020] Zheng, W., Wang, K., and Wang, F. Y. (2020). A novel background subtraction algorithm based on parallel vision and Bayesian GANs. *Neurocomputing*, 394:178–200.

[Zhong et al., 2019] Zhong, Y., Ji, P., Wang, J., Dai, Y., and Li, H. (2019). Unsupervised deep epipolar flow for stationary or dynamic scenes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 12087–12096.

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

[Zhou et al., 2020] Zhou, W., Deng, Y., Peng, B., Liang, D., and Kaneko, S. (2020). Co-occurrence background model with superpixels for robust background initialization. *arXiv report arXiv:2003.12931*.

[Zhu et al., 2021] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.

[Zhuang et al., 2019] Zhuang, C., Zhai, A., and Yamins, D. (2019). Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 6001–6011.

List of Figures

1.1 Examples of real-time traffic webcam image available on the Internet 8

2.1 Example of state of state of the art unsupervised unstructured representation learning model using image completion as pretext task: MAE 14
2.2 Overview of the SimCLR model 15
2.3 Transformer architecture 19
2.4 Example of class activation maps 20
2.5 Example of multi-heads attention maps extracted from the last layer of a self-supervised VIT-S model 21
2.6 Example of scene graph 23
2.7 Example of application of a Nerf model 24
2.8 Example of image warping using a spatial transformer network 25
2.9 Overview of the SPACE model 26
2.10 hierarchical latents in StyleGAN 30
2.11 Overview of the ALOE model 31
2.12 Example ALOE VQA predictions 32
2.13 Example of future frame prediction of the ODDN model . . 32

3.1 Schematic of loss function and gradient computation 39
3.2 Overview of the stochastic gradient descent optimization process 42
3.3 Examples of background reconstruction 49
3.4 Examples of background reconstruction 50

4.1 examples of predictions of the proposed model 52
4.2 Schematic of the proposed model during inference 57
4.3 Examples of background reconstruction and foreground segmentation 59

4.4 Examples of background reconstruction and foreground segmentations on the datasets Clevrtex, ObjectsRoom and ShapeStacks . . . 62
4.5 Failure cases due to overfitting on the datasets CDnet 2014 and BMC 2012 . . . 64
4.6 Examples of background reconstruction and foreground segmentation on the CDnet 2014 dataset . . . 68
4.7 Examples of background reconstruction and foreground segmentation on the CDnet 2014 dataset . . . 69
4.8 Examples of background reconstruction and foreground segmentation on the LASIESTA dataset . . . 70
4.9 Examples of background reconstruction and foreground segmentation on the BMC 2012 dataset . . . 71
4.10 Examples of background reconstruction and foreground segmentation on Clevrtex dataset. . . . 72
4.11 Examples of background reconstruction and foreground segmentation on ObjectsRoom dataset. . . . 72
4.12 Examples of background reconstruction and foreground segmentation on ShapeStacks dataset. . . . 73

5.1 Overview of the KNEEL model . . . 78
5.2 Overview of proposed model . . . 79
5.3 Examples of segmentation predictions on CLEVRTEX, CLEVR, ShapeStacks, ObjectsRoom, OOD and CAMO test datasets 86
5.4 Examples of segmentation predictions on CLEVRTEX, CLEVR, OOD and CAMO test datasets obtained using other models 87
5.5 Examples of glimpses . . . 88
5.6 t-SNE plots of the distribution of the z_{what} vectors associated to positive detections on the ObjectRoom and CLEVRTEX datasets . . . 89
5.7 Examples of segmentation predictions on real-world traffic videos . . . 90
5.8 Examples of object glimpses generated from real-world traffic videos and associated input images, reconstructed images and predicted segmentation maps . . . 91
5.9 t-SNE plots of the distribution of the z_{what} vectors associated to positive detections on two real-world traffic videos . 92
5.10 t-SNE plot of the distribution of the z_{what} vectors associated to positive detections on one real-world traffic video . 93
5.11 Examples of segmentation predictions on CLEVR test dataset 99
5.12 Examples of segmentation predictions on CLEVRTEX test dataset . . . 100

5.13 Examples of segmentation predictions on ObjectsRoom test dataset 101
5.14 Examples of segmentation predictions on ShapeStacks test dataset (using a model without transformer) 102
5.15 Examples of segmentation predictions on CAMO test dataset using a model trained on CLEVRTEX only 103
5.16 Examples of segmentation predictions on OOD test dataset using a model trained on CLEVRTEX only 104
5.17 Examples of segmentation predictions on a real-world video extracted from a traffic webcam 105
5.18 Examples of segmentation predictions on a real-world video extracted from a traffic webcam 106
5.19 Examples of segmentation predictions on a real-world video extracted from a traffic webcam 107

6.1 Overview of the uORF model, which uses conditional NeRFs as object and background generators 110
6.2 Overview of the model proposed in [Ye et al., 2022], which uses deformable sprites as object and background generators 110

List of Tables

3.1 Evaluation results per criteria on the SBMnet 2016 dataset 44

3.2 Evaluation results for the AGE criterion per category on the SBMnet 2016 dataset 44

3.3 Evaluation results per criteria on the SBI dataset 45

3.4 AGE scores obtained using various truncated versions of the algorithm 46

3.5 Impact of reducing the number of iterations 47

4.1 Comparison of top BGS algorithms according to the per-category F-measures on CDnet-2014 60

4.2 Average per category of video F-measures on LASIESTA . . 61

4.3 Comparison of top unsupervised BGS algorithms according to the video F-measure on BMC 2012 61

4.4 F-Measure on the Clevrtex, ShapeStacks and ObjectsRoom datasets 62

4.5 F-measure results obtained on the CDnet dataset with a model pretrained using the first half of each video as training set, and fine-tuned on the last half using various numbers of fine-tuning iterations 63

4.6 Computation time of the proposed model, PAWCS and SubSENSE for some sequences of the CDnet and BMC datasets 64

4.7 Evaluation of various ablations of the proposed model . . . 65

4.8 Number of channels for each layer of the encoder and decoder (excluding positional encoding input channels) . . . 66

4.9 autoencoder architecture for 64×64 images 67

4.10 autoencoder architecture for 128×128 images 67

5.1 Benchmark results of the unsupervised object-centric representation model on CLEVR and ClevrTex 84

5.2 Benchmark results on ObjectsRoom and ShapeStacks . . . 84

5.3 Benchmark generalization results on CAMO, and OOD for a model trained on CLEVRTEX 85
5.4 Results of ablation study and additional experiments 88
5.5 Training computation time with one Nvidia RTX 3090 GPU 94
5.6 Hyperparameter values 95
5.7 glimpse generator architecture 97
5.8 U-net architecture (ablation study) 98

RÉSUMÉ

L'objectif de cette thèse est d'étudier comment les techniques d'apprentissage profond, c'est-à-dire la descente de gradient stochastique et les réseaux de neurones, peuvent être utilisées pour obtenir une représentation interprétable d'une scène sans nécessiter de jeu de données annotées. Afin d'obtenir une telle représentation, nous considérons qu'une scène est composée d'un arrière plan et de divers objets apparaissant en avant-plan. Nous devons donc non seulement être capable de distinguer l'arrière-plan de ces différents objets, mais aussi de séparer ces objets, qui peuvent se toucher ou s'occulter entre eux.

Nous étudions d'abord la tâche de reconstruction d'arrière-plan fixe, dont le but est de construire une image unique de l'arrière-plan d'une scène à l'aide d'une courte séquence d'images de cette scène encombrée par divers objets. Nous considérons cette tâche comme un problème d'estimation robuste, proposons une nouvelle technique appelée bootstrap d'arrière-plan, qui utilise la descente de gradient stochastique, et montrons qu'elle est plus précise et considérablement plus rapide que les meilleures méthodes existantes.

Nous considérons ensuite la tâche de reconstruction d'arrière-plan dynamique et de segmentation d'arrière-plan/avant-plan. À partir de l'hypothèse selon laquelle les arrière-plans des images apparaissant dans une vidéo ou un jeu de données sont situés sur une variété de petite dimension, nous sommes en mesure d'apprendre cette variété à l'aide d'un autoencodeur convolutionnel. Afin d'améliorer les résultats de segmentation, nous adaptons l'autoencodeur pour prédire le bruit d'arrière-plan, qui peut être causé par la turbulence ou les mouvements des arbres ou de l'eau. Nous montrons ensuite que le modèle proposé donne de meilleurs résultats que les meilleures méthodes non supervisées existantes sur les exigeants benchmarks CDnet et LASIESTA.

La segmentation de l'arrière-plan est une première étape pour comprendre la structure d'une scène, mais elle ne permet pas d'identifier et de segmenter les divers objets apparaissant dans une scène. Afin d'obtenir une représentation véritablement centrée sur les objets d'une scène, nous introduisons une nouvelle architecture pour l'apprentissage non supervisé de représentations centrées sur les objets, qui utilise l'attention et le soft-argmax pour localiser chaque objet et un transformer encodeur pour gérer les occlusions et éviter les doubles détections. Nous montrons ensuite que cette architecture est considérablement plus précise que l'état de l'art sur les benchmarks synthétiques existants et fournissons quelques exemples d'applications à des images réelles prises par des caméras de circulation.

MOTS CLÉS

représentation structurée, représentation centrée sur les objets, segmentation d'objets, detection d'objets, arrière-plan, avant-plan, apprentissage non supervisé

ABSTRACT

The goal of this thesis is to study how deep learning techniques, i.e. stochastic gradient descent and neural networks, can be used to get an interpretable representation of a scene without requiring any annotated dataset. In order to get such a representation, we consider that a scene is composed of a background and various foreground objects. We then have to be able to distinguish the background from the foreground objects present in the scene, and also to separate these foreground objects, which can touch or occlude each other.

We first study the task of fixed background reconstruction, whose goal is to build a unique background image of a scene using a short sequence of images of this scene cluttered by various objects. We address this task as a robust estimation problem, propose a new technique called background bootstrapping, which uses stochastic gradient descent, and show that it is more accurate and significantly faster than state of the art methods.

We then consider the task of dynamic background reconstruction and background/foreground segmentation. Starting from the assumption that the backgrounds of the images appearing in a video or a dataset lie on a low dimensional manifold, we are able to learn this manifold using a convolutional autoencoder. In order to improve segmentation results, we adapt the autoencoder to predict the background noise, which can be caused by turbulence, moving trees or water, and should not be considered as foreground. We then show that the proposed model is able to improve upon the state of the art for unsupervised methods on the challenging CDnet and LASIESTA benchmarks.

The segmentation of the background is a first step in order to understand the structure of a scene, but it does not allow to identify and segment the various objects appearing in a scene. In order to get a true object-centric representation of a scene, we introduce a new architecture for unsupervised object-centric representation learning, which uses attention and soft-argmax to localize each object and a transformer encoder to manage occlusions and avoid duplicate detections. We then show that this architecture is significantly more accurate than the state of the art on existing synthetic benchmarks and provide some examples of applications to real-world images taken from traffic cameras.

KEYWORDS

structured representation, object-centric representation, object detection, object segmentation, background, foreground, unsupervised learning