Robust Visual Features for Long-Term Monitoring

Assia Benbihi

CentraleSupélec THALES IAEM Georgia Tech cnrs
Unité Mixte Internationale 2958

Robust Visual Features for Long-Term Monitoring

THÈSE DE DOCTORAT

(mention Informatique)

présentée et soutenue en visio-conférence en le 22 Mai 2020

par

Assia Benbihi

CentraleSupelec

Thales SIX GTS

Ecole Doctorale IAEM

Informatique, Automatique, Electronique Electrotechnique, Mathématiques

Laboratoire UMI-2958 GeorgiaTech CNRS

Composition du jury

Directeur de thèse: :	Cédric Pradalier
Co-directeur de thèse: :	Matthieu Geist
Président du jury :	Cédric Demonceaux
Rapporteurs :	Cédric Demonceaux Torsten Sattler
Examinateurs :	Margarita Chli Matthieu Geist Cédric Pradalier
Invité :	Laurent Frerebeau

NNT : 2020CSUP0002

Résumé

L'inspection visuelle consiste à observer une scène et mesurer ses changements. Parmi ses nombreuses applications figurent la conduite autonome, l'inspection industrielle ou encore la réalité augmentée. La principale difficulté pour une machine consiste à reconnaitre une scène malgré que cette dernière puisse changer d'apparence. C'est sur cette problématique que se concentre cette thèse et plus particulièrement sur la reconnaissance à long terme de scènes bucoliques, comme la rive d'un lac au fil des saisons. Le but est ensuite de quantifier les variations de cette scène. L'approche adoptée se divise en deux étapes : la reconnaissance de la scène puis la mise en correspondance de zones locales de celle-ci.

La reconnaissance visuelle se base sur des représentations de l'image telles que deux images d'une même scène ont des descriptions similaires, et ce même lorsque l'apparence visuelle de la scène a changé. Une des contributions majeures de cette thèse est la définition de deux descripteurs d'image reposant sur la géométrie et la sémantique de la scène. Etant donnée que ces deux informations sont globalement invariantes au temps, les descripteurs le sont également. Comparés à l'état de l'art, ces descripteurs sont parmi les plus performants pour la reconnaissance de scènes bucoliques et généralisent même aux scènes urbaines.

Pour mettre en correspondance des zones de deux images d'un même scène, ce sont leurs descriptions locales qui doivent être invariantes. L'approche adoptée dans cette thèse est d'identifier les zones locales de l'image qui restent invariantes en exploitant la structure de l'espace image de réseaux de neurones déjà entrainés sur une tâche visuelle quelconque. Les représentations locales qui en résultent sont tout aussi pertinentes pour la mise en correspondance de zones d'image que celles issues d'un apprentissage profond spécifiquement dédié à cette tâche.

Enfin, cette thèse introduit deux méthodes d'apprentissage visant à réduire le volume de données nécessaires à l'entrainement de réseaux de neurones pour la segmentation sémantique. En plus de pouvoir s'intégrer à la description d'image, les information sémantiques permettent de rendre d'autres applications visuelles plus robustes aux changement d'apparence. Ceci est illustré par un exemple appliqué au cas de l'odométrie visuelle directe.

Abstract

Visual monitoring consists of observing a scene and tracking its modifications. This task is integrated into most of autonomous systems relying on localization such as autonomous driving, industrial inspection or augmented reality. One of the main challenges is to define a robust image representation that allows an autonomous system to recognize a scene even when its appearance changes. Given images of the same scene, the goal is then to characterize the scene's variations over time. This thesis addresses this challenge in bucolic environments over long periods such as parks or lakeshores across seasons. The adopted approach is to first put the images to compare in correspondence and then measure their variations.

The image correspondence problem is split into two sub-problems: scene recognition and image local matching. In scene recognition, the challenge is to define an image representation such that corresponding images have similar descriptions even when there are strong variations in visual appearance. One major contribution of this thesis is the definition of two novel image descriptors based on the geometry and semantics of the scene. Since these two properties are mostly invariant over time, the resulting descriptors are also invariant. Experiments on two bucolic environments show that they reach state-of-the-art performance. They are also compared with deep learning approaches on urban scenes whereas they do not need training.

For image matching, the challenge is similar but at the scale of image regions. The problem consists in choosing image regions relevant to the monitoring task and generating a representation invariant to appearance variations. The second major contribution of this thesis is to query such regions and representations from a neural network. A trained network generates a powerful representation space and experiments show that it reaches the same matching performances as networks specifically trained for image matching.

The remaining contribution of this thesis studies how to reduce the training load to obtain efficient semantics. They are not only useful to define new image representation but they also make existing localization approaches more robust. For example, this thesis shows that it improves the tracking robustness in direct visual odometry.

To my parents.

"These rocks remind me of the 12 Apostles."
- Cédric, JNRR 2017, Biarritz, France.

"Indeed."
- Assia, ICONIP 2019, 12 Apostles, Australia.

Acknowledgements

Three years ago, I started a ~~Ph.D.~~ POMDP. As the ~~Ph.D. student~~ agent, I had access to a noisy observation of the state~~-of-the-art~~ and had to select the next research action to take. It could result in a contribution that updated the state, with a transition probability that neared randomness. The convergence of my ~~research~~ decision process heavily depended on my ~~advisors~~ reward functions. They provide the agent with an incredible amount of both scientific and human knowledge, experience, and lifetime lessons. Today, I am converging to one optimal Ph.D. policy thanks to their guidance through this highly uncertain environment, and I will be forever humbled and grateful to Cédric Pradalier and Matthieu Geist.

Besides, I want to deeply thank Laurent Frerebeau for enabling this Ph.D. through Thales and offering me three years of research freedom. I thank Suzel Lavagne for challenging me with near-impossible engineering puzzles, and Pascal Delalande, Sébastien Le Stum and Kévin Jezequel for the short but intense scripting initiation that saved me precious experimenting time throughout my Ph.D. I also thank the sweet Anne Viguie for her heartwarming welcomes, and the whole SLS team for bearing with my debug venting and for playing rubber duck whenever I got stuck,

I want to thank my colleagues from the DREAM lab and especially Xiaolong for initiating me to direct visual odometry, Antoine and Stephanie for the endless philosophical discussions, Antoine for hosting the game nights, Georges for teaching us to the best food places, Othmane for motivating me to the Monday workouts, and Laura for the sweets that made the workout worth it. I also thank Nadège for the forever useful information.

Finally, I thank my parents for teaching me the priceless value of hard work and perseverance.

Thank you, Australia for welcoming me during this writing period.

Contents

Chapter 1

Introduction

Visual monitoring consists of tracking a scene's variations over time. When integrated into autonomous systems, it allows for safe and cost-efficient monitoring of remote and hazardous areas, and facilitate systematic environment assessment over long periods. Visual monitoring relies on four major visual tasks: scene recognition, scene localization, scene alignment, and scene comparison. Each of these tasks is also of pivotal importance in other domains. For example, mobile systems usually rely on visual localization to get finer camera pose than the one provided by a GPS.

The visual primitive common to all of these tasks are visual features. A feature is a compact, informative and discriminative representation of the image content. For localization applications, an ideal feature should depend only on the image content and not on the image appearance. This means that it should be invariant to variations such as illuminations or viewpoints. However, most of the existing features are built upon the statistics of the image pixels and not upon the image content. So, whenever the image's appearance changes, these features do too, even when the image content does not. This can falsely lead an autonomous system to think that the scene changed or that it is observing a different scene when it is not the case. This problem is a major challenge in computer vision research and this motivates this thesis to introduce visual features with improved invariance properties.

The rest of this section is organized as follows. Section (Sec.) 1.1 illustrates the necessity of visual features invariant to image appearance changes. Sec. 1.2 lists the contribution of this thesis towards robust visual features. Sec. 1.3 summarises this thesis and outlines its structure.

1.1 Robust Features for Visual Monitoring

This section depicts visual use cases for which robust features are crucial.

Figure 1.1: The Sydney Opera House pictured with two extremely different light conditions and opposite viewpoints.

Global image features. A global image feature is a representation vector that describes the whole image content. Ideally, a robust visual feature should be invariant to a wide range of appearance changes including illumination, viewpoint, season and weather. Depending on the nature of the environment to monitor, some of these requirements can be disregarded. For example, most images taken in cities depict buildings that are naturally invariant to seasons. So the robustness to seasonal variations is less critical. Currently, the main challenges for urban scene recognition is to define a global image descriptor invariant to wide viewpoint variations and extreme illumination changes. Figure (Fig.) 1.1 shows an example with the famous Sydney Opera House taken from opposite viewpoints and times of the day. Existing visual features are not invariant to such color and geometry variations, which motivates most of the literature to focus on these two invariance axes. One reason that explains this research bias for urban settings is the high amount of public datasets available, which allows for a wider range of approaches,

including machine learning. Conversely, this thesis addresses the robustness problem for bucolic environments such as forests or lakeshores. There, the nature of the variations requires visual features to be robust to a wider range of variations, which leads to a different approach for image description.

Figure 1.2: Bucolic scenes exhibit additional types of variations compared to urban ones. For example, the vegetation can grow with time.

In addition to the illumination and the viewpoint variations, visual features for bucolic scenes must also be robust to weather and seasons. Even in the ideal case where two images are taken from the same viewpoint and under the same light conditions, the vegetation state can induce differences in the image. The koala bush in Fig. 1.2 is a good example: the picture on the left was taken during a sunny spring day and the koala looks green and well-fed. Several months later, the koala looks grey, probably because of the hot summer weather. It also seems to have lost a few pounds and let its fur grow. A robust visual feature should be invariant to these changes and describes the koala structure the same way throughout the year.

A major contribution of this thesis is the definition of two global image descriptors based on the geometry and the semantics of the scene. As these elements are mostly invariant over time and seasons, the resulting features also exhibit robustness.

Local image features. The previous example discussed global image description, *i.e.*, one representation to describe the image. Another category of visual features describes local regions of the image. These local features should exhibit the same robustness as their global counterparts but also handle low-level visual distractors such as poor image texture or repetitive patterns. An ideal method should select image regions consistent over several images, and provide invariant descriptions over appearance variations. Thus, image regions with the same content should have similar local features.

Figure 1.3: Challenging local matching over bucolic environments. Left: The camera displacement is obvious to the human eye and to the camera as the image holds several salient and unique structures such as the clock. Right: The camera rotation and translation are less perceivable.

As with the global approach, most of the literature focuses on illumination and wide-viewpoint variations over urban structures. The release of such large datasets has especially allowed deep learning approaches to tackle the robustness problem. Most of the approaches optimize a model on a specific type of data to detect and describe image regions. The main advantage is that features are data-specific so they should inherently be robust to the data appearance variations. However, this comes at the cost of computationally demanding optimization and time consuming human supervision. This motivates this thesis to define data-specific features while reducing the training overhead.

Another major contribution of this thesis is to leverage the powerful representation space generated by a trained Convolutional Neural Network (CNN) to define data-specific local features. The CNN can be queried for relevant image location by computing the gradient of the feature map norm with respect to the image. This derivation outputs a saliency map of which local maxima are the image regions to describe. The same network is used to describe these local regions by interpolating the CNN's features on these locations. Experiments show that such local features are as performant as when the neural network is specifically trained for feature detection.

Semantics to robustify visual odometry. Semantics is one of the few visual information that stays invariant over time. It cannot only be leveraged to define new localization algorithms but also integrated into existing ones to make them more robust to image variations. This is illustrated with the case of edge-based direct visual odometry, for which the use of semantic edges instead of standard ones improves its tracking robustness.

Alternatives to Human Supervision for Semantics. Another relevant application for semantics is the assessment of the scene's variations. Figure 1.4 shows two aerial images of Wallabi Point, Australia, before and after the bushfires. A pixel-wise semantic segmentation of these images allows for fast and automatic assessment of the fires' impact in the region. However, current segmentation methods require heavy human supervision to be robust to such appearance variations. So this thesis proposes two approaches to reduce the supervision load while maintaining the segmentation performance.

Figure 1.4: A useful application of autonomous visual monitoring is the automatic assessment of the landscape changes after a catastrophe, such as the 2019 Australian bushfires [212]. Supervised segmentation can currently provide pixel-wise labeling of the land across these changes but requires heavy training with human annotation.

1.2 Contributions

The contribution of this thesis is the definition of robust visual features for visual monitoring that require few or no human supervision at all. It is divided into four parts, each one addressing one of the challenges described in the previous section.

The first part introduces two novel global image descriptors robust to season variations and suitable for the recognition of bucolic scenes. The second part describes how to design data-specific local features without supervision by leveraging a CNN's representation space. The third part illustrates how integrating semantics into existing

edge-based visual odometry algorithms can increase their tracking robustness. The last part introduces alternatives to human supervision to adapt segmentation across image domains.

This thesis builds on the following articles:

> [25] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Geometric and semantic visual words for scene recognition across seasons. *submitted*, 2019
>
> [23] Assia Benbihi, Stéphanie Arravechia, Matthieu Geist, and Cédric Pradalier. Image-based place recognition on bucolic environment across seasons from semantic edge description. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020
>
> [24] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Elf: Embedded localisation of features in pre-trained CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7940–7949, 2019
>
> [27] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Semi-supervised domain adaptation with representation learning for semantic segmentation across time. In *International Conference on Neural Information Processing*, pages 459–466. Springer, 2019
>
> [195] Xiaolong Wu, Assia Benbihi, Antoine Richard, and Cédric Pradalier. Semantic nearest neighbor fields monocular edge visual-odometry. *arXiv preprint arXiv: 1904.00738*, 2019
>
> [148] Antoine Richard, Assia Benbihi, Cédric Pradalier, Vincent Perez, Philippe Durand, and Rosalinde Van Couwenberghe. Automated segmentation and classification of land use from overhead imagery. In *International Conference on Precision Agriculture*, 2018

The contributions in [148] and [195] are the results of collaborations with my Ph.D. colleagues Antoine Richard and Xiaolong Wu where I participated in the experiments and in the writing. In [148], I contributed to the finetuning of CNNs for semantic segmentation of land occupation over aerial images. In [195], I collected and generated the data necessary to run the semantic-edge-based visual odometry optimization.

The following publication is not included in the manuscript as it addresses another aspect of visual monitoring on the sensor placement problem and to learn it from expert's demonstrations.

[26] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Learning sensor placement from demonstration for UAV networks. In *Proceedings of the IEEE Symposium on Computers and Communications*, 2019

1.3 Overview

This section summarizes the thesis.

Part I: Robust visual features for scene recognition. Part one addresses the problem of scene recognition in bucolic environments across seasons. Here, scene recognition is framed as image retrieval where a query image is matched to the most similar image available in a database. The search is computed on representations with a much lower-dimensional space than the image. The challenge is to compute a compact image encoding such that images of the same location are near to each other despite their change of appearance due to environmental changes.

Chapter (Chap.) 3 [23] introduces a global image descriptor computed from semantics and geometry. By embedding these two image invariants, this descriptor exhibits robustness against variations in visual appearance such as illumination, vegetation state, weather, and viewpoint. It is built from the wavelet transforms of the image's semantic edges. Matching two images amounts to matching their semantic edge descriptors. This method reaches State-of-the-Art (SoA) performance for image retrieval on two multi-season environment-monitoring datasets: the Extended-CMU-Seasons [157] and the Symphony Lake [66] datasets. It also generalizes to urban scenes on which it is on par with the current baselines NetVLAD [9] and DELF [133].

While this representation exhibits higher robustness to seasonal changes than the SoA, it has several limits. The first one is that it only leverages global geometric information instead of the local edge variations. Another major drawback is the lack of scalability with the number of edges. Finally, matching two scenes requires solving the linear assignment problem between edge descriptors over the whole database. In **Chap. 4** [25], a better global descriptor addresses these limits while maintaining the scene recognition performances. It aggregates hand-crafted local geometric features with semantic constraints into a compact vector. The local features are acceleration maxima of semantic edges, described with the edges' local variations. They are aggregated into label-specific visual words in a semantic VLAD fashion.

Part 2: data-specific features without supervision. Part 2 tackles the challenge of defining local features that are data-specific without the heavy supervision needed to reach SoA. To do so, **Chap. 5** [24] introduces a novel feature detector based only on information embedded inside a CNN already trained on standard learning tasks, such as classification, with no further training. Keypoints are the local maxima of the gradient of the feature maps' norm with respect to the image. Local descriptors are generated by interpolating one of the CNN feature maps on these keypoints locations. Contrary to recent CNN-based methods, this method requires neither training nor finetuning, except on the task it was previously trained on. When compared with hand-crafted or learning methods, it reaches the same performance in terms of repeatability and matching score on the HPatches and Webcam datasets. It also compares to their robustness against light, scale, rotation and viewpoint changes. This shows that the feature representation and localization information learned by a CNN to complete a visual task is as relevant as when the CNN is specifically trained for feature detection.

Part 3: Pixel-Wise semantics for Robust localization. Part 3 illustrates how integrating pixel-wise semantics into edge-based visual odometry makes the visual tracking robust. **Chap. 6** [195] extends existing edge-based visual odometry with the additional constraint that associated edges should have the same semantic class. When tested on the KITTI and vKITTI dataset, experiments show that it not only reaches lower trajectory error but also exhibits a larger basin of attraction during the tracking phase. This latter property makes it more robust to viewpoints variations due to large camera displacements.

As for the semantic-based global descriptors of **Part 1** [23, 25], the main performance bottleneck is the segmentation accuracy. Currently, segmentation reaches top accuracy only when heavily trained with pixel-wise annotations. This motivates the last part of this thesis to investigate alternative supervision to train segmentation.

Part 4: Alternative Supervision for Segmentation Domain Adaptation. **Chap. 7** [148] proposes to transform existing annotated datasets into the desired domain using neural style transfer. Style transfer is the task of reproducing an image content with the style of another image. The output image keeps the same content but exhibits a pixel distribution nearer to the style image. This chapter relies on this concept to transform existing annotated images towards the target pixel distribution. It is tested to segment overhead land images where each pixel is classified with land types. In this example, annotations are available only for images sampled in 2015, but not for images from previous years.

The pixel distribution changes over time mostly because the data collection process is updated. So even though they depict a similar scene, a CNN trained on the 2015 images cannot generalize to the previous years. So the goal is to transform the 2015 images to match the distributions of the past years and use these synthetic images to train a network. Experiments show that the CNN still needs real annotated images to converge but these synthetic images prove relevant to warm-up the network. Warming-up the network can reduce the amount of real data needed for the finetuning, and even provide a better initialization state that boosts the segmentation performance.

Although it does not require human supervision, the previous domain adaptation by neural style transfer requires heavy computations. To alleviate the expensive data generation process, **Chap. 8** [27] proposes a semi-supervised method for the specific case of images with similar semantic content but different pixel distributions. This is a common occurrence in long-term monitoring tasks. Given a network trained with supervision on a past dataset, a copy of this network is finetuned on the new dataset to preserve its features maps. The domain adaptation becomes a simple regression between the past and new feature maps and does not require annotations on the new dataset. In other words, the CNN is trained to project two images with the same content but different appearance onto the same point in the CNN's representation space. This method reaches performances similar to classic transfer learning on the PASCAL VOC dataset whereas it does not require additional supervision. The domain changes are emulated with synthetic transformations such as color histogram variations, edge noise, and texture artifacts.

Chapter 2

Related Work

Those who cannot remember the past are condemned to repeat it.
- George Santayana

This chapter reviews the literature relevant to this thesis's contributions. Sec. 2.1 introduces the key concepts it builds upon such as projective geometry, local and global features, and saliency. Then, Sec. 2.2 discusses the main research on local features. They characterize local regions of the image and their aggregation provides image summaries relevant to scene recognition reviewed in Sec. 2.3. Finally, Sec. 2.4 presents experimental contributions researching the correlations between the representation space of a CNN and the image space.

2.1 Concepts

The reader familiar with projective geometry, visual features, and CNN may skip this section and start reading Sec. 2.2.

This section briefly reviews the camera model and the relation between the 3D world and the image plane. This is useful to understand the contribution in Chap. 6 on Visual Odometry (VO). Given two images on the same scene with different viewpoints, VO aims at recovering the camera pose using geometric constraints between the images, the 3D information of the scene and pixel correspondences across images. Two pixels correspond when they project to the same 3D point in the scene.

In Chap. 6, the camera pose and the 3D information are jointly and iteratively adjusted until the corresponding pixels project to the same point in the scene. This optimization is guided by the projection error and is dubbed *direct VO*. *Indirect VO* differs in that it relies on previously computed pixel matches to constrain an equation system where the unknowns are the camera pose parameters. The pixel pairings are computed

beforehand based on their visual similarity: each pixel is assigned with a *descriptor*, *i.e.*, a compact representation of its neighborhood. When combined with the pixel position, it forms a *local feature* that characterizes the pixel. Chap. 5 introduces a novel approach to compute such features.

These same local features are also used to compute a global summary of images, relevant for scene recognition addressed in Chap. 3 and Chap. 4. Given an image database, it searches for the entry that is the most similar to a query image. Rather than comparing the images, it compares their compact summaries, also called *global features*.

Visual features are also leveraged for higher-level visual tasks such as object classification or semantic segmentation. In general, these features can be either hand-crafted or learned. While there is a wide literature on machine learning for image processing [28], recent approaches mostly rely on a specific set of learning tools: Convolutional Neural Networks. The last part of this section provides a brief introduction to convolutions (see [63] for a deeper description).

Camera Model.

This subsection introduces a simple camera model based on the projection of 3D points onto a plane. The projection equations are useful to better grasp the visual odometry contribution in Chap. 6.

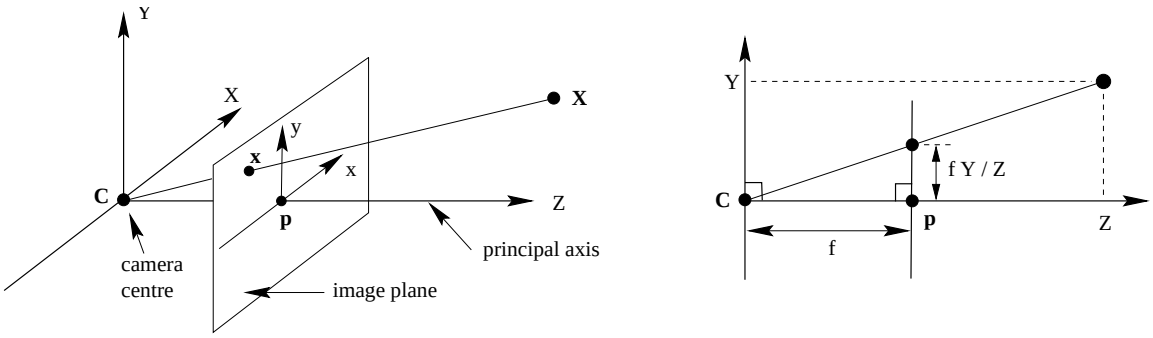


Figure 2.1: Pinhole camera model illustration [72]. All coordinates are expressed in the camera frame centered in C with the camera looking down its Z axis. The image plane is defined by the equation Z = f with f a parameter called the *focal distance*. $\boldsymbol{X}$ is a 3D point in the scene projected to a point $\boldsymbol{x}$ the image plane using Thales's theorem.

An image is the projection of the 3D world onto a plane. The projection is achieved by letting the world's light rays go through a small opening (*e.g.* a diaphragm) and focusing them on a plane using lens optics. Fig. 2.1 illustrates a simplified projection using the simple *pinhole* camera model. Here, the lens is discarded and it is assumed that the

camera opening is an ideal point, so one light ray going from X is projected on the image plane. More complex models can be found in [72] but they are not required for following the rest of this thesis.

The 3D world is described in the camera frame whose origin is the camera center $\boldsymbol{C}$. The axes are defined such that the camera looks down the Z axis and the (X,Y,Z) basis is orthonormal. The image plane is defined by the equation Z = f, with f being a parameter called the *focal distance*. The plane's (x,y) axes are colinear to the (X,Y) ones. Its origin is the point $\boldsymbol{p} = (0,0,f)^{\mathrm{T}}$ defined as the intersection of the Z axis with the plane. A 3D point in the camera frame is noted $\boldsymbol{X}_{\mathrm{C}} = (x,y,z)^{\mathrm{T}}_{\mathrm{C}}$. Its projection $\boldsymbol{x}$ is defined as the intersection of that plane with the line ($\boldsymbol{X}_{\mathrm{C}}$C). The coordinate of the projection are computed with the Thales's theorem $\boldsymbol{x}_{\mathrm{C}} = (\frac{x \cdot f}{z}, \frac{y \cdot f}{z})^{\mathrm{T}}$ in the camera frame (Fig. 2.1 - right). In the rest of this thesis, the points on the image plane (*i.e.*, the pixels) will always be expressed in the camera frame and the subscript $\boldsymbol{x}_{\mathrm{C}}$ will be omitted to write $\boldsymbol{x}$.

Homogeneous coordinates are introduced to write this projection as a linear transformation. They are equivalent to the previous cartesian coordinates and one can go from one form to the other using the following relations (Equation (Eq.) 2.1- 2.2). For a point in the scene, let $\boldsymbol{X}$ and $\boldsymbol{x}$ be its cartesian coordinates of the 3D point and its projection. Their homogeneous counterparts, noted $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{x}}$, are defined as follows with $x, y, z, u, v, w \in \mathbb{R}$:

$$\boldsymbol{X} \triangleq \begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \bar{\boldsymbol{X}} \triangleq \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \qquad \boldsymbol{x} \triangleq \begin{pmatrix} u \\ v \end{pmatrix} \rightarrow \bar{\boldsymbol{x}} \triangleq \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{2.1}$$

The conversion from homogeneous coordinates is computed as follow:

$$\bar{\boldsymbol{X}} \triangleq \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \rightarrow \boldsymbol{X} = \begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \\ \frac{z}{w} \end{pmatrix} \qquad \bar{\boldsymbol{x}} \triangleq \begin{pmatrix} u \\ v \\ w \end{pmatrix} \rightarrow \boldsymbol{x} = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix} \tag{2.2}$$

Projecting the 3D point onto the image plane can now be written in matrix form by introducing a matrix $\boldsymbol{K}$, called the intrinsic matrix:

$$
\begin{aligned}
\mathrm{K} &\triangleq \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
\text{and} \quad \bar{\boldsymbol{x}} &= \begin{pmatrix} \boldsymbol{K} & \boldsymbol{0} \end{pmatrix} \bar{\boldsymbol{X}} \quad \text{with} \quad \boldsymbol{0} = {}^{\mathrm{T}}(0,0,0) \\
\begin{pmatrix} u \\ v \\ w \end{pmatrix} &= \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}
\end{aligned}
\tag{2.3}
$$

Note that the intrinsic matrix $\boldsymbol{K}$ in Eq. 2.3 has a simple form only because it is associated with the pinhole camera model. Other camera models have additional non-zero terms to better approximate the projection. This thesis does not address these additional terms and will note the projection matrix $\boldsymbol{K}$.

Until now, the coordinates of the scene point $\boldsymbol{X}$ were defined relative to the camera frame. This is not convenient as they will change every time the camera moves. Instead, it is better to express $\boldsymbol{X}$'s coordinates relative to the world frame and transform them into the camera frame at the time of the projection. This is achieved by updating Eq. 2.3 into Eq. 2.4 to compute $\boldsymbol{X}$'s coordinates in the camera frame before the projection. Let SO(3) be the orthogonal group of dimension 3, ${}^{\mathrm{C}}\boldsymbol{R}_{\mathrm{W}} \in \mathrm{SO}(3)$ be the rotation and ${}^{\mathrm{C}}t_{\mathrm{W}}$ the translation that transform from world to camera coordinates. The transformation matrix from the world frame to the camera frame is called the *extrinsic* matrix and is noted ${}^{\mathrm{C}}\mathrm{T}_{\mathrm{W}} \in \mathbb{R}^{4\times4}$. The scene point expressed in the world frame is noted $\bar{\boldsymbol{X}}_{\mathrm{W}}$. The projection formula becomes:

$$
\begin{aligned}
{}^{\mathrm{C}}\mathrm{T}_{\mathrm{W}} &\triangleq \begin{pmatrix} {}^{c}\mathrm{R}_{\mathrm{W}} & {}^{\mathrm{C}}t_{\mathrm{W}} \\ 0 & 1 \end{pmatrix} \\
\text{and} \quad \bar{\boldsymbol{x}} &= \boldsymbol{K}\,{}^{\mathrm{C}}\mathrm{T}_{\mathrm{W}}\bar{\boldsymbol{X}}_{\mathrm{W}} \\
\begin{pmatrix} u \\ v \\ w \end{pmatrix} &= \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} {}^{\mathrm{C}}\mathrm{R}_{\mathrm{W}} & {}^{\mathrm{C}}t_{\mathrm{W}} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{\mathrm{W}}
\end{aligned}
\tag{2.4}
$$

It is possible to project the pixel $\boldsymbol{x} = (u, v)^{\mathrm{T}}$ back to the 3D world $\boldsymbol{X}_{\mathrm{W}}$, given that the pixel's depth z is available:

$$
\begin{aligned}
\bar{\boldsymbol{X}}_{\mathrm{W}} &= {}^{w}\mathrm{T}_{c}\begin{pmatrix} z \cdot \boldsymbol{K}^{-1}\bar{\boldsymbol{x}} \\ 1 \end{pmatrix}_{\mathrm{W}} \\
\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}_{\mathrm{W}} &= \begin{pmatrix} {}^{c}\mathrm{R}_{\mathrm{W}}^{\mathrm{T}} & -{}^{c}\mathrm{R}_{\mathrm{W}}^{\mathrm{T}} \cdot {}^{c}t_{\mathrm{W}} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} z \cdot \begin{pmatrix} \frac{1}{f} & 0 & 0 \\ 0 & \frac{1}{f} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \\ 1 \end{pmatrix}
\end{aligned} \tag{2.5}
$$

Visual odometry.

This subsection describes the two main camera pose derivations in visual odometry. Once again, the curious reader can refer to [72] for further details.

Visual odometry is the task of recovering the pose of the camera from image information only. Given two images depicting the same scene from different viewpoints, it leverages pixel associations and the geometric constraints between the image planes, the camera poses, and the scene depth to recover the camera displacement.

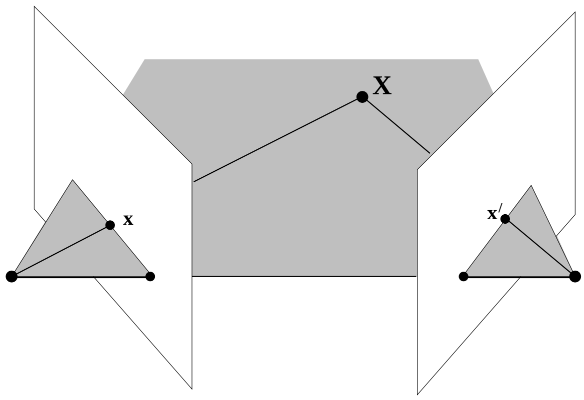


Figure 2.2: From [72]. $\boldsymbol{X}$ is a 3D point in the scene observed by two cameras. It projects onto the pixels $\mathbf{x}$ and $\mathbf{x}'$. The pixel coordinates, the camera displacement and the z coordinate of $\boldsymbol{X}$ in the world frame are related. Visual odometry leverages this correlation to recover the camera displacement from pixel pairs. Direct approaches use these pairs to enable their iterative optimization whereas indirect approaches use them to constrain equation-based approaches.

More formally, let C and C′ be two cameras with intrinsic matrices K_{C} and $\mathrm{K}_{\mathrm{C}'}$ capturing the same scene with two different viewpoints, resulting in images I_{C} and $\mathrm{I}_{\mathrm{C}'}$. There can be one camera only moving around in which case $\mathrm{K}_{\mathrm{C}} = \mathrm{K}_{\mathrm{C}'}$. Let ${}^{\mathrm{C}'}\mathrm{R}_{\mathrm{C}}$ (respectively (resp.) ${}^{\mathrm{C}'}t_{\mathrm{C}}$) be the rotation (resp. the translation) from C′ to C. The goal is to recover ${}^{\mathrm{C}'}\mathrm{R}_{\mathrm{C}}$ and ${}^{\mathrm{C}'}t_{\mathrm{C}}$. There are two broad categories of visual odometry: direct method and indirect ones, also called feature-based ones.

Direct methods. Direct approaches iteratively adjust ${}^{C'}R_C$ and ${}^{C'}t_C$, and the depth of the scene to best align I_C over $I_{C'}$. In Fig. 2.2, this amounts to finding the camera displacement and the depth of X to best project $\boldsymbol{x}$ over $\boldsymbol{x}'$, and vice-versa. When the correspondence between $\boldsymbol{x}$ and $\boldsymbol{x}'$ is known, the projection is assessed with the distance between $\boldsymbol{x}'$ and the projection $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ (Fig. 2.3). The analog distance is computed for the reverse projection. However, in practice, this is not the case. So an alternative solution is to compare the pixel intensity at $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}'$. With the assumption that the illumination stays the same across the two images. the projection is correct when $I_{C'}[\hat{\boldsymbol{x}}] = I_{C'}[\boldsymbol{x}]$.

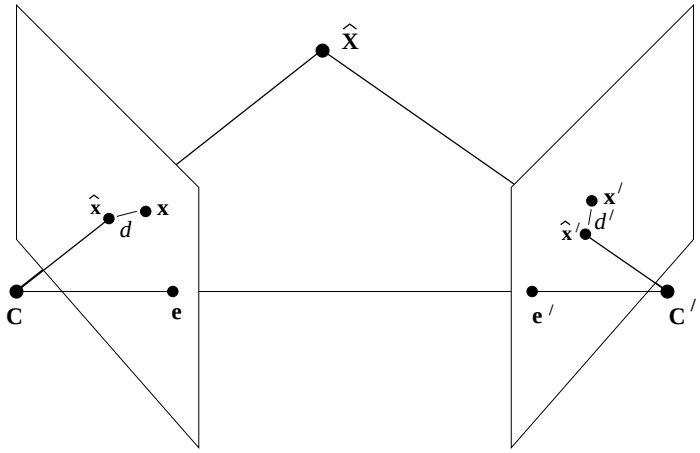


Figure 2.3: From [72]. $\hat{\boldsymbol{X}}$ is a 3D point in the scene with an estimated depth $\hat{z}$ with respect to the camera frame $\mathbf{C}$. It projects to $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$. The actual 3D point $\boldsymbol{X}$ with the correct depth projects onto the pixels $\mathbf{x}$ and $\mathbf{x}'$. The reprojection error penalizes the camera displacement and the depth estimation with the distance d (resp. d') between the projected pixel $\hat{\mathbf{x}}$ (resp. $\hat{\mathbf{x}}'$) with the projection target $\mathbf{x}$ (resp. $\mathbf{x}'$).

More formally, each pixel $\boldsymbol{x} = (u, v)^{\mathrm{T}}$ in I_C has an estimated depth z with respect to the camera frame. It is projected onto a pixel $\hat{\boldsymbol{x}} = \pi(\boldsymbol{x})$ (Eq. 2.6). This equation projects $\boldsymbol{x}$ on the 3D scene with the projection matrix K_C and the estimated depth. The resulting 3D point's coordinates are relative to C's frame. They are transformed into C′'s frame using the estimated camera displacement. The 3D point is projected on C′'s image plane using its projection matrix $K_{C'}$. Finally, the homogeneous coordinates are converted into cartesian ones.

$$\pi(\mathbf{p}) = \begin{pmatrix} \frac{\tilde{u}}{\tilde{w}} \\ \frac{\tilde{v}}{\tilde{w}} \end{pmatrix} \quad \text{with} \quad \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = K_{C'} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} {}^{C'}R_C & {}^{C'}t_C \\ 0 & 1 \end{pmatrix} \begin{pmatrix} K_C^{-1} \begin{pmatrix} z \cdot u \\ z \cdot v \\ z \end{pmatrix} \\ 1 \end{pmatrix} \tag{2.6}$$

Assuming that the illumination stays constant across the images, the alignment is evaluated with the intensity variation of the projected pixel between $\mathrm{I_C}$ and $\mathrm{I_{C'}}$. More formally, the loss to minimize is:

$$\mathscr{L} := \sum_{\mathbf{p} \in \mathrm{I_C}} \|\mathrm{I_{C'}}[\pi(\boldsymbol{p})] - \mathrm{I_C}[\boldsymbol{p}]\|_2 \tag{2.7}$$

In practice, the loss accumulates the intensity difference only for pixels which projection falls in the second image. This sum of squared difference is minimized with standard optimization. This leads to satisfying results as long as the pixel intensity does not change across images. This assumption is called the *brightness* consistency assumption and is fairly realistic in indoor environments where the light can be controlled. However, it is easily violated in outdoor settings for which indirect methods are better suited.

Indirect methods. Indirect methods, also called feature-based methods, assume that a set of pixel correspondences $(\boldsymbol{x}, \boldsymbol{x'})$ is available. This allows constraining a system of geometric equations where the unknowns are the camera displacement terms. The derivation of the pixel correspondences is the object of the next subsection on local features.

The geometric constraint is called the *epipolar constraint*. It is derived from the frame transformation between the two camera frames (Eq. 2.8). Let $\boldsymbol{X}_\mathrm{W}$ be a 3D point expressed in the world frame and $\boldsymbol{X}_\mathrm{C}$ and $\boldsymbol{X}_{\mathrm{C}'}$ its coordinates expressed in the camera frames C and C′. Let $\boldsymbol{x}_\mathrm{C}$ and $\boldsymbol{x}_{\mathrm{C}'}$ be the pixels where this points project on each image plane. The epipolar constraint relates these pixels with relation $\boldsymbol{x}_{\mathrm{C}'}^\mathrm{T} \boldsymbol{F} \boldsymbol{x}_\mathrm{C} = 0$, where $\boldsymbol{F}$ is called the *fundamental matrix*. It is derived as follows:

$$\begin{aligned}
\boldsymbol{X}_{\mathrm{C}'} &= {}^{\mathrm{C}'}\mathrm{R_C} \cdot \boldsymbol{X}_\mathrm{C} + {}^{\mathrm{C}'}t_\mathrm{C} \\
\Rightarrow \quad {}^{\mathrm{C}'}t_\mathrm{C} \times \boldsymbol{X}_{\mathrm{C}'} &= {}^{\mathrm{C}'}t_\mathrm{C} \times ({}^{\mathrm{C}'}\mathrm{R_C} \cdot \boldsymbol{X}_\mathrm{C} + {}^{\mathrm{C}'}t_\mathrm{C}) \\
\Rightarrow \quad \boldsymbol{X}_{\mathrm{C}'}^\mathrm{T} \cdot ({}^{\mathrm{C}'}t_\mathrm{C} \times \boldsymbol{X}_{\mathrm{C}'}) &= \boldsymbol{X}_{\mathrm{C}'}^\mathrm{T} \cdot ({}^{\mathrm{C}'}t_\mathrm{C} \times {}^{\mathrm{C}'}\mathrm{R_C} \cdot \boldsymbol{X}_\mathrm{C}) \\
\Rightarrow \quad 0 &= \boldsymbol{X}_{\mathrm{C}'}^\mathrm{T} \cdot ({}^{\mathrm{C}'}t_\mathrm{C} \times {}^{\mathrm{C}'}\mathrm{R_C} \cdot \boldsymbol{X}_\mathrm{C}) \\
\Rightarrow \quad 0 &= (\mathrm{K_{C'}^{-1}} \boldsymbol{x}_{\mathrm{C}'})^\mathrm{T} \cdot ({}^{\mathrm{C}'}t_\mathrm{C} \times {}^{\mathrm{C}'}\mathrm{R_C} \cdot (\mathrm{K_C^{-1}} \boldsymbol{x}_\mathrm{C})) \\
\Rightarrow \quad 0 &= \boldsymbol{x}_{\mathrm{C}'}^\mathrm{T} (\mathrm{K_{C'}^{-1}})^\mathrm{T} \cdot ({}^{\mathrm{C}'}t_\mathrm{C} \times {}^{\mathrm{C}'}\mathrm{R_C} \cdot (\mathrm{K_C^{-1}} \boldsymbol{x}_\mathrm{C})) \\
\\
\Rightarrow \quad 0 &= \boldsymbol{x}_{\mathrm{C}'}^\mathrm{T} \mathrm{F} \boldsymbol{x}_\mathrm{C} \\
\text{with} \quad \mathrm{F} &= (\mathrm{K_{C'}^{-1}})^\mathrm{T} \cdot ({}^{\mathrm{C}'}t_\mathrm{C} \times {}^{\mathrm{C}'}\mathrm{R_C}) \cdot \mathrm{K_C^{-1}}
\end{aligned} \tag{2.8}$$

The fundamental matrix has seven degrees of freedom (a 3×3 has eight independent ratios and F satisfies the additional constraint $\det \mathrm{F} = 0$, which removes one degree of freedom). Seven epipolar constraints are enough to recover F *i.e.* although, in practice, it is better to use more to account for the possible imprecision and errors in the pixel correspondence $(\boldsymbol{x}, \boldsymbol{x}')$. Once F is computed, it is possible to recover ${}^{\mathrm{C}'}t_{\mathrm{C}}$ and ${}^{\mathrm{C}'}\mathrm{R}_{\mathrm{C}}$ from it.

Local Features.

This subsection provides an intuitive idea of what local features are. See Sec. 2.2 for an in-depth review.

Figure 2.4: Keypoints are pixels that are easily matchable across images. In this example, the center of the green circles are keypoints matched across two viewpoints.

One way to compute corresponding pixels between two images is to select the most characteristic regions common to both, and try to associate them. Similarly to direct odometry, pixels should be matched when they are the projection of the same scene point. However, without prior geometric knowledge, only the image visual content is left to compute such pairs. So two pixels should match when they point to similar visual elements (*e.g.*, the green circles in Fig. 2.4). This is enabled with *Local features* that are compact summaries of the visual content around a pixel. It is then possible to efficiently compute the similarity between pixels and associate the nearest ones with each other.

A local feature is made of the pixel coordinates and a vector representing the image content around it, called a *descriptor*. A good description is one that characterizes uniquely each image region with the same vector across images. Thus, the local feature can be consistently matched based on their descriptors. In the example Fig. 2.4, the description of the clock center should be the same in the left and the right image. However, it should be significantly different from the description of the plaque under it.

A standard way to compute descriptors is to rely on high-level statistics of the neighboring pixels. These are usually invariant to pixel intensity changes so they stay consistent over images taken under various light conditions. So pixel correspondences can be computed even when the brightness consistency assumption is violated. This is why indirect methods are currently preferred in outdoor environments. Still, robustifying direct approaches is the object of recent encouraging contributions [52, 196, 197, 213, 214].

There are mainly two levers to define ideal features: the pixel description and the pixel selection. A perfect descriptor should be map all the patches pointing to the same element of the scene to the same descriptor. This condition is extremely hard to comply with and is not even necessary to derive the fundamental matrix, which is the motivation for pixel correspondences. Indeed, the derivation can theoretically be achieved with at least seven pairs of matching pixels. So this condition is simplified by selecting only a subset of characteristic pixels in the two images and associate them. Feature detection is the problem of selecting the most relevant pixels to describe and match.

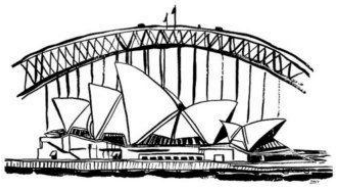
Image Retrieval.

The local features previously introduced can be used for more than pixel association. When fused, they generate a global summary of the image that is relevant for image retrieval applications, of which the Google Image browser is the most popular. Fig. 2.5 illustrates an example: the images at the top are stored in a database with their global descriptions. Given a query image (bottom), the browser computes a query descriptor and retrieves the database images with the nearest description. The main challenge is to compute image summaries that are robust to variations in the image appearance such as day/night, ground/aerial views, or season changes. This is complex because it requires the features to ignore the pixel variations related to these changes and only capture the high-level content of the scene. CNN are relevant tools to compute such features and their basic operations are described in the next subsection.

Convolutional Neural Networks (CNN).

Convolution. A convolution is a linear operator between two functions. In signal processing, it is how filters are mathematically represented. In image processing, the functions are discrete and defined over 2D spaces. A convolution is defined by $(f * g)(u_0, v_0) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u, v) \cdot g(u_0 - u, v_0 - v)$, with f and g in $\mathbb{R}^{\mathbb{R}^2}$.

Database images.



Query image.

Figure 2.5: Retrieval example. Top: Various images of the Opera House stored in a database along with other pictures. Bottom: Query image. Image retrieval searches for the database images most similar to the query one.

Convolutions allow computing image statistics such as gradients or edges. In the previous definition, f is a *convolution filter* or *kernel*, and g is an image. The filter usually has a finite support so the two infinite sums become finite. More specifically, let $\mathrm{I} \in \mathbb{R}^{h \times w}$ be an image of height $h \in \mathbb{N}$ and width $w \in \mathbb{N}$, and $\mathrm{W} \in \mathbb{R}^{k \times k}$ be a filter of size $k \in \mathbb{N}$, with $k \equiv 1[2]$. For each pixel $(u, v) \in \mathbb{N}^2$, the convolution outputs a real value computed as:

$$\begin{array}{rccl} \mathrm{W} * \mathrm{I}: & \mathbb{R}^2 & \to & \mathbb{R} \\ & (u_0, v_0) & \to & \displaystyle\sum_{u=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{v=-\frac{k-1}{2}}^{\frac{k-1}{2}} \mathrm{W}[u, v] \cdot \mathrm{I}[u_0 - u, v_0 - v] \end{array} \tag{2.9}$$

Figure 2.6 illustrates the operation on a small example with a 3×3 convolution kernel (gray square). The blue square is a 5×5 one channel image and the green square is the convolution output. The illustration is borrowed from the insightful report of Dumoulin

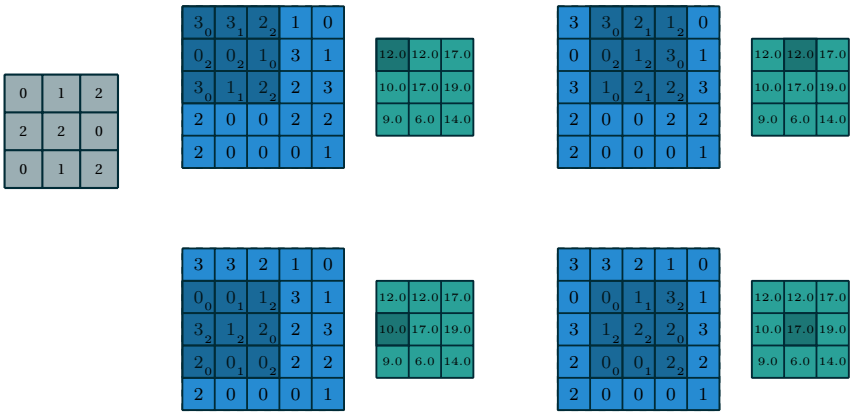


Figure 2.6: Illustration borrowed from the insightful convolution guide from Dumoulin and Visin [50]. Discrete convolutions are computed by sliding the convolution kernel (grey) over the image, and summing the output of the element-wise multiplication. Light blue: one-channel image. Dark blue: kernel aligned over the image. Light green: convolution output. Dark green: current convolution output.

and Visin on image convolutions [50]. The convolution is computed by sliding the kernel over the image and accumulating the element-wise multiplication. A classic example is the Sobel filter that computes the first order derivative of the pixel intensity along the x and y axis (Fig. 2.7). The output of each convolution is a map with high activations where the intensity gradients are high. Since edges are image areas where the pixel intensity varies, these convolutions are useful to find image edges.

CNNs. Instead of hand-crafting the filter weights, it may be easier to learn them. For example, one can learn the Sobel filter by replacing the convolution values with unknowns $w_{i,j}$ (Fig. 2.8). Given a set of images with their respective Sobel outputs, the filter weights are iteratively updated until they generate the expected Sobel output. Note that this assumes there exist examples of images with their Sobel outputs.

The Sobel filters are quite straightforward and learning them seems like overkill. However, the composition of a high number of these simple filters can also be trained to learn visual information as complex as object classification [98, 171], semantic segmentation [18, 36, 210], image saliency [137, 101], image features [24, 46, 51, 135, 199].

A *filter bank* is a set of convolutional filters applied to an input image. The outputs of these filters are concatenated to form a *feature map*. This map can also be the input to further filters and the outputs can be fused in various ways such as averaged, concatenated, or summed. The composition of successive filters allows deriving pixel statistics

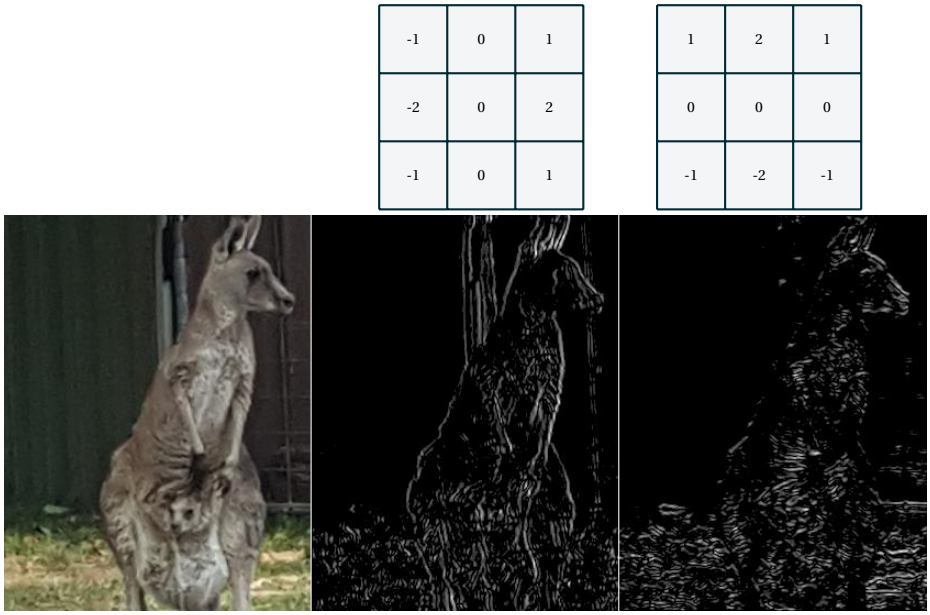


Figure 2.7: Left-Right: Image, Vertical Sobel filter, Horizontal Sobel filter. The filters compute the first order derivative of the pixel intensity along the y and x axis. It highlight the edges, *i.e.*, an area where the pixel intensity varies.

$w_{0,0}$	$w_{0,1}$	$w_{0,2}$
$w_{1,0}$	$w_{1,1}$	$w_{1,2}$
$w_{2,0}$	$w_{2,1}$	$w_{2,2}$

Figure 2.8: Parametric convolution with learnable weights $w_{i,j}$.

at such a high-level that low-level intensity variations become negligible. This is what makes CNN features relevant for robust image description.

2.2 Local features

A local feature is defined by a keypoint and its descriptor. A *keypoint* is a location in the image, specified by image coordinates or an area. It can be augmented with the local orientation of the pixels or the scale of the image local content. For example, image corners are suitable keypoints [71]. A *descriptor* is a vector that characterizes the local region around the keypoint in the most discriminative way possible. An example is the histogram of the gradient's orientation of the neighboring pixels [116]. In an ideal world, there would exist a bijective mapping between the set of all image patches depicting the

same scene structure and a descriptor. This would allow the unique identification of point in the scene from its descriptor only, no matter the appearance of the image patch. One way to achieve this is to define descriptors with infinite dimensions, but this is not tractable. Instead, an easier constraint to comply with is to make the descriptor bijective only over the set of selected keypoints.

This constraint sets the quest for the holy grail of features. Good keypoints form a set over which a bijective descriptor can be defined. And a good descriptor should be a bijection over the selected keypoints. See how the noble quest has now turned into a chicken and egg problem? Should we start by defining a descriptor function and then select keypoints over which it is bijective? Or should we first select keypoints and then define the descriptor?

Rather than fall into this never-ending loop, the literature has agreed on some properties for good features. They should be *repeatable*: this means that the selected locations should be consistent over the scene even when the illumination or the viewpoint change. They should be easy to *match*: this is achieved by selecting discriminant image locations, which are easier to characterize uniquely. There can be additional constraints depending on the final application. When used in localization, for example, the keypoint locations are better specified with sub-pixel precision. Else, the camera pose derived for the matches can lack accuracy. In image retrieval, this constraint can be slightly lessened when computing the image description, but no more than to a few pixels precision. However, the post-processing requires the same precision as for the localization.

The literature on local features is extremely rich and too large to be entirely reviewed here. So, this section focuses on reviewing the history of local features rather than describing the list of all the contributions. Extensive evaluations of the main contributions described in this section are available in [106, 130, 161].

Applications. Figure 2.9 illustrates an example of two images of the same street. The local features are matched according to their descriptor distance: each feature from the first image is matched to the feature with the most similar descriptor in the second image. This generates a list of corresponding pixels between the two images.

Assuming that the features are 'good', the absence of correspondences is usually a reliable indication that the images do not depict the same scene. Given enough correspondences, one can recover the transformation between the two images. The derived transformation is used to project the features from the first image onto the other. The quality of the transformation is measured with the number of inliers, *i.e.*, the number of features from the first image that are projected near their matching features on the

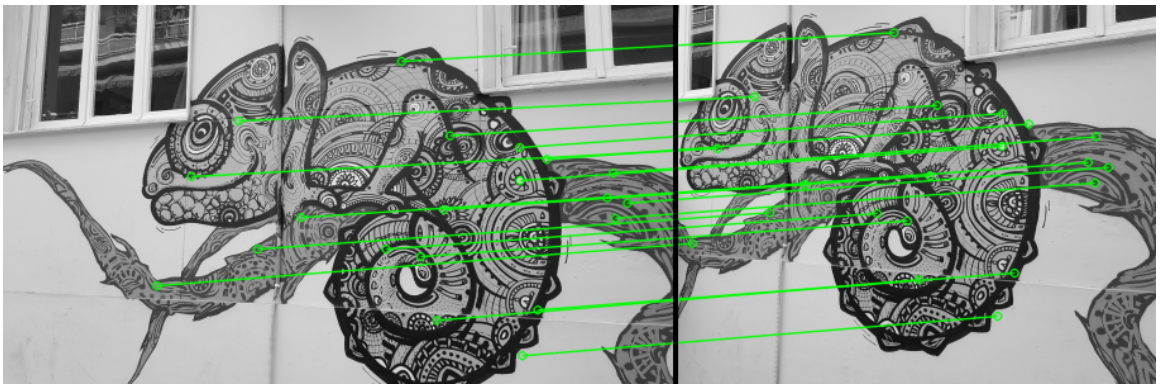


Figure 2.9: Two images depicting the same street art in Athens with different viewpoints. The green dots locate the features. The green lines link the matching ones. Only a small subset of matches is drawn for visibility purposes.

second image. A low ratio of inliers is also a reliable indication that the images do not depict the same scene. Note that this post-processing is reliable only if the local features perform well.

The previous processing is called *geometric verification* and is one of the many applications for local features. It is a standard way to improve the performance of image retrieval [142, 172] by checking that the retrieved images are geometrically coherent with the query. Given a query image and a retrieved one, few inliers indicate that these two images do not depict the same scene. Since pictures are rarely taken under the same illumination or viewpoint conditions, robust features are essential. They must detect consistent points across a wide range of scene variations to allow reliable matching. The description must also ignore appearance changes and only embed the invariant content around the keypoint location.

Another popular application is Structure-from-Motion (SfM) [160]. Given a set of images depicting a scene from various viewpoints, it reconstructs the 3D representation of the scene by estimating the camera pose and the scene's depth. The first step of SfM is the derivation of local features. These are matched over several images based on their descriptor distances. The resulting feature pairs are used to constrain the camera poses and the depth. First, a subset of the 3D structure is reconstructed from a pair of images. Then, SfM alternates between estimating the pose of the next image to integrate and the depth of its local features. The whole pipeline relies on local feature matching so this application is a good example of the importance of robust features.

Another type of localization is visual Simultaneous Localization And Mapping (SLAM) [44, 131]. An autonomous system must reconstruct an unknown environment and localize itself inside it. The first step is called the *mapping* and the second the *tracking*. One

way is to sample images during the exploration and, once again, compute pixel correspondences between successive images. This generates geometric constraints used to estimate the camera displacements and sparse depth of the scene. As the robot goes, it integrates these successive displacements and the 3D information into a trajectory localized in the 3D scene.

Hand-crafted Local Features. Early methods rely on hand-crafted detection and description. This means the criteria to select keypoints are manually set, as well as the pixel statistics to describe it.

One of the most popular detectors is the Harris Corner [71]. The idea relies on the intuition that corners are distinctive and repeatable in the image. So keypoints are pixels belonging to corners and they store the corner's scale with respect to the image scale. In practice, a corner is characterized by strong variations of the image gradients in two directions. This phenomenon is quantified by the Harris corner function: it compares the amplitude of the gradient variations over two orthogonal axes. If only one is high, the pixel probably lies on an edge and not a corner. If both are high, the pixel lies on a corner. These two amplitudes are computed as the eigenvalues of the Hessian matrix in the second-order approximation of the gradient variations around the keypoint.

While corners are robust to illumination and viewpoint variations, they are not consistent when the image scale changes. This motivates [124] to integrate automatic scale selection [111, 193] to define the Harris-Laplacian detector. It looks for corners over multiple scales and keeps only the most prominent ones. The bottleneck of this approach is the duration of the Hessian computation. Bay *et al.* [21] address this problem with the Speeded-Up Robust Features (SURF) feature. Given an image, they compute the image integrals [192] to reduce the Hessian computation to only 3 additions. Another of their contribution is a descriptor defined as a 2D Haar wavelet transform of the keypoint region. This amounts to decomposing the image patch over a finite basis of patterns.

While SURF benefits from the Wavelet invariance properties, the gold standard for local features stays SIFT [115, 116]. The first difference with the previous approach is that SIFT detects invariant blobs rather than corners. A formal approach to do so is to select maxima of Laplacians of Gaussians of the image. But a faster computation based on the Difference of Gaussians (DoG) of the image is used in practice. The blobs are detected over multiple scales of the image. The resulting keypoints are image pixels with the scale at which they are found. The keypoint location is estimated with precision even lower than the pixel by interpolating the DoG signal both in scale and in space. As motivated previously, such a resolution is crucial for localization applications.

The SIFT descriptor is computed from the pixel statistics over a circular region around this keypoint and proportional to its scale. First, the canonical orientation for the patch is estimated by the direction with the highest cumulative intensity gradient. The patch is then divided into sub-patches and for each sub-patch the orientations of the pixel gradients are accumulated in a histogram. The descriptor is the concatenation of all the sub-histograms. In practice, SIFT has proved its robustness over the decades against a wide range of datasets. In theory, the robustness to scale is explained by the multi-scale detection and scale adaptive description. The robustness to orientation is provided by the patch rectification to its canonical orientation. And the robustness to illumination is enforced by the use of image gradients for the detection and the description. Gradient is usually invariant to global intensity changes, such as the ones caused by a variation in illumination.

Another approach similar to blob detection is introduced in [121]. The Maximally Stable Extremal Region (MSER) detector segment image areas that are the most invariant to affine transformations of pixel intensity. Formally, given an intensity threshold, it segments contiguous areas which intensity is higher (or lower) than the threshold and that stay invariant to a finite range of such thresholds. This design has the advantage of intrinsically detecting image areas that are robust to illumination variations.

While the previous approach, and more specifically SIFT, offer robustness against a wide range of variations, they were previously deemed too slow and memory-greedy for real-time applications such as SLAM [44] or autonomous navigation [42]. This motivated part of the research effort to tackle efficiency in addition to robustness. FAST [151] proposes to detect corners using an efficient pixel test. A pixel is a corner only if its intensity is lower (or brighter) than most of the pixels located on a circle around it. The authors note that the test can be speeded up by selecting only a subset of these neighboring pixels. The subset and the order of pixels to test is optimized to minimize the number of pixels needed to correctly classify a pixel as a corner or not. They derive a classification tree where each node is a pixel of the circle and keep only the most informative one for the binary decision. In a way, FAST is one of the first learning-based detectors. This detector is fast enough for real-time application, and its extension AGAST [118] is even faster. Another extension improves the pixel selection strategy [152]. But the main limit of FAST is that it is not tailored to handle scale variations in the images.

The ORB feature [153] tackles this issue by running the FAST detector on multiple scales of the same image. It also augments the BRIEF [32] descriptor to make it robust to rotation changes. Given an image patch, the BRIEF descriptor aggregates the responses

to manually defined binary tests between pairs of pixels. An example is to sample randomly 128 pairs of pixels and test whether the intensity of the first is higher than the second. The main advantage of such a descriptor is that it is binary. So during the feature matching, the descriptor distance can be a simple bit comparison. This is especially suitable for robotics and other real-time applications. However, the ORB's authors observe that BRIEF is sensitive to in-plane rotations. So they compute the canonical patch orientation and rotate the pixels before aggregating the test binary responses.

The BRISK [108] feature also builds on the FAST and BRIEF approaches. It runs FAST on multiples scale of the image and generates a FAST score map. As in SIFT, keypoints are local extrema of these score maps over both image and scale spaces. The descriptor is analog to BRIEF in that it aggregates binary responses of pixel comparisons. However, they sample the pixels according to a circular pattern around the keypoint, in a similar fashion to the Daisy descriptor [184]. The difference with [184] is that they sample fewer pixels to keep the descriptor more compact. [7] also investigates the best pixel pattern to sample the binary tests.

KAZE [8] breaks with the quest for computational efficiency and tackles the problem of the scale space derivation. The standard way to derive the scale space of an image uses the Gaussian kernel but this induces a loss of resolution that hinders the precision of the keypoint localization, and then the performance of the target application. One solution already introduced in SIFT is to interpolate the keypoint location in the scale space. Instead, KAZE proposes to derive the scale space with other filters that better preserve the localization information. It relies on existing filtering work from [194] to derive a data-adaptive scale space where blurring is reduced over locations likely to hold keypoints, such as object boundaries. This allows for better keypoints localization, which is crucial for most of the applications. Similar work can be found in [35, 139].

Even though these hand-crafted methods have proven to be successful, recent approaches prefer to rely on machine learning and more specifically deep learning.

Detector Learning. One of the first detectors trained for robustness to drastic natural illumination variations is TILDE [191]. It is optimized to select consistent keypoints on the Webcam dataset that depicts six static outdoor scenes over a wide range of natural illumination and weather, *e.g.*, sunset, sunrise, night, overcast, fog. Inspired by hand-crafted approaches, the model generates a score map in which local maxima are the selected keypoints. The regression model is supervised so that the local maxima occur at the locations of SIFT robust keypoints. A SIFT keypoint is robust if it is consistent over at least 100 images of the same scene.

TILDE is one of the most robust detectors on the Webcam dataset but one drawback is that it relies on the hand-crafted SIFT for supervision. This biases the keypoint selection towards what the SIFT criterion would already select. A solution is to train a model without supervision as in [158]. The authors train a neural network to rank keypoints according to their robustness to random hand-crafted transformations. Then they keep the top/bottom quantile of the ranking as keypoints. Here, the supervision signal is the robustness of the keypoints. Analog work is proposed in [107, 209, 207] where the general idea is to keep pixels that are consistent over local image transformations.

Recent experiments show that CNN can be a relevant source of keypoint locations. The CNN's feature maps were previously deemed too coarse to carry local information. But the experimental study of Long *et al.* [113] suggests that features correlate to local image regions at a lower resolution than their large receptive field. This work is one of the motivations of the contribution in Chap. 5: ELF [24] extracts localization information for the feature maps and also uses these maps to define the descriptor. The resulting features are as performant as when the CNN is specifically trained to detect features. Following work [51, 166] comforts the assumption that the feature space of a trained CNN embeds relevant information for keypoint detection.

Although unsupervised approaches present the advantage of breaking with human bias, current SoA is reached either by SIFT, heavily supervised methods or recent approaches that leverage CNN representation space.

Descriptor Learning. Even though recent research favors the joint training of detector and descriptor, there is significant literature on individual descriptor learning. Most of it follows this general pipeline: train a neural network to generate compact descriptors of the image regions so that these representations are close if the content is similar, far otherwise. Papers differ in the network architecture they use, the definition of descriptor similarity, whether they work on image patches or the whole image, and the training loss.

The first deep descriptors rely on Siamese networks [39] fed with image patches. They are trained so that the descriptors of matching patches stand close to each other. A standard loss is the contrastive loss [68]: it minimizes the distance between matching descriptors and maximizes it when they do not. Following works investigate the choice of the training loss: DeepDesc [168] and [204] use a hinge loss between the two descriptors, [73] uses the mean Average Precision metric as a loss, Matchnet proposes to learn both the descriptor network and the similarity network [69]. In addition to minimiz-

ing the descriptor distance, L2-net [180] also constrains the network's feature maps for matching patches to be close.

Instead of comparing only pairs of descriptors, [20, 100, 209] use a triplet loss: three patches are fed to the same network, two of them match and the third does not. The triplet loss forces the distance between the non-matching descriptor to be higher than the distance between the matching ones by a certain margin. Inspired by Lowe's ratio, [127] minimizes the distance between matching patches but then differs by maximizing the distance between the patch and his second nearest neighbor. [91] points out the lack of constraint of the feature distance as an issue. In practice, most of the models converge but this has raised questions on how to efficiently chose negative examples. Investigating how to mine these negative examples is a recurrent problem [127, 146, 147].

Rather than generating descriptors, [200] proposes to learn a post-processing step that classifies matching pairs as correct or not. Another line of work rather focuses on the network architecture to make the descriptors robust rather than on the loss or the data: [56] introduces a pooling method to make descriptors rotation invariant. Similarly, [40] integrates a spatial transformer network [78] in their architecture.

In the detector learning literature, experiments from [113] suggest that the representation space generated by a trained CNN embeds information on keypoints at a satisfying resolution. At the same time, [57] shows that the feature space of a CNN is also discriminative enough to extract local descriptors from it. Their results show that such descriptors even compare to SIFT and can be used for other standard vision tasks [164].

Even though CNNs are the most widespread learning models in computer vision, other models are also relevant. [189] defines a descriptor as a weighted combination of image responses and the combination is optimized using boosting. [170] augments the SIFT Histogram of Oriented Gradients (HoG) descriptor by learning a pooling pattern over it and a reduction step.

The current standard for local descriptor training is 'Universal Correspondences Network (UCN)' [40]. Given a pair of matching images, the network generates a feature map for each. For each keypoint in the image, a 3d voxel is extracted from the feature map that corresponds to that point location. The network is trained with the correspondence contrastive loss: the voxel distance between matching keypoints should be minimized and the one between non-matching points should be higher than a margin m. It is integrated into the recent SoA end-to-end learning methods SuperPoint [46]. Another efficient loss is the triplet loss that contrains the difference between matching and non-matching distance to be higher than a margin m. Examples comprising this loss include [20, 127] and the recent D2-Net [51].

End-to-end Learning. Instead of separately training detection and description, recent research favors joint or end-to-end learning.

One pioneer of this approach is LIFT [199]. The method comprises three CNNs, one for each of the standard steps: keypoint selection, estimation of the canonical orientation, and keypoint description. They first start with the description training as in [168]: the network is fed with a triplet of patches with two of them matching and the third one is a negative example. The descriptor is trained to produce similar descriptors for the matching patches and a distinct one for the negative patch. They rely on SfM to generate pairs of matching patches: once the 3D environment is reconstructed, they sample 3D points and crop the images around the corresponding 2D image points. The trained descriptor is then used to supervise the orientation network [128] on pairs of matching matches. This second network should output the patch orientation that minimizes the descriptor distance between the corresponding patches. Finally, the detector is integrated with the two previous networks and trained. Given an image, it outputs a score map in which local maxima are the keypoints' location. The three networks are jointly optimized so that matching detected keypoints have close descriptors, non-matching ones have distinct descriptors and non-keypoint locations are not detected.

The same authors later propose an optimization to train the detector, the orientation estimator and the descriptor jointly [135]. They rely on images for which the depth and the camera extrinsics are available, or rely on SfM to generate it. The descriptor network and the optimization are similar to LIFT's ones. Again, the detector network outputs a score in which local maxima are the keypoints location. The first difference lies in the multi-scale processing: they resize the detector's feature map before the output at multiple resolutions. This generates score maps at multiple scales. These are then resized to the image resolution and fused. The same feature map is leveraged to estimate image orientation at each pixel by feeding to a distinct convolutional filter bank. This is more efficient than the multiple feeds in LIFT. The detector is trained on two images depicting the same scene and for which the camera extrinsic and the depth are provided. This allows warping the first image over the other, *i.e.*, project the pixels of the first image onto the second. The detector is trained to generate consistent score maps over the two images. This is measured by how well the score map of the first image is aligned with the second's one after warping. In practice, the first score map of the image is post-processed before the warping. The detector is also constrained to output the same orientation and scale for the detected keypoints, *i.e.*, the score map maxima. LF-Net almost doubles the performance of LIFT even when they are trained on the same

data. This suggests that although LF-Net benefits from more training data, this novel joint optimization is beneficial.

LF-Net reaches SoA performance on urban outdoor scenes but SuperPoint [46] is slightly more successful on indoor scenes. One of the main strengths of SuperPoint is its self-supervised approach and the high amount of data it was trained on. It first trains a detector on synthetic shapes to detect shape junctions as keypoints: this teaches the network that corners are patterns. The detector classifies each pixel as junction or not at a lower resolution than the image. The loss is a simple cross-entropy loss. The network is adapted to natural images on the MS-COCO [110] dataset. Once again, self-supervision is used: for each image, they sample pseudo-ground-truth keypoints and warp the image with random homographies. They then train the detector to detect the warped keypoints on the warped image. A key element to the CNN performance is the homography sampling (details in [46]). Once the detector converges, they train the descriptor using the previous work from [40]. The network generates feature maps for a pair of images. For each keypoint in the image, a 3d voxel is extracted from the feature map at the point location. The network is trained so that the distance between these voxels is minimized for matching keypoints, and maximized otherwise. Using synthetic homographies tells whether the keypoint match or not.

Recently, the philosophy supporting better exploitation of CNNs representation space has regained interest [57, 113]. D2-Net [51] leverages the localization information present in the trained CNN feature map to detect keypoints. The CNN is trained with the triplet margin loss. One of the feature maps is selected to extract keypoints from. Keypoints are pixels that are both local spatial maxima over one channel and maxima along the feature channels. Rather than defining an individual detection loss, they only extend the description one by weighting the triplet loss. The weight encourages the network to generate high feature scores for locations that are well described and improve the features for corresponding local maxima that have poor descriptors.

Although most of the end-to-end learning methods push the SoA in terms of matching scores, they require a significant amount of data and complex optimization. In analogy with the past efforts to make hand-crafted features more efficient, this thesis tackles the problem of reducing the training overhead while maintaining the data-specific of these features [26]. This is the contribution of Chap. 5 that relies on CNN already trained on a standard vision task. It then extracts the relevant information already embedded inside the network for local feature detection, which requires no training nor supervision. A score map is generated from the gradient of the feature map norm with respect

to the image in one backward pass. A standard Non Maxima Suppression (NMS) samples the local maxima as keypoints. One of the CNN feature map is interpolated at the keypoint locations to generate descriptor vectors.

2.3 Global Features

A global image feature is a descriptor that characterizes the whole image rather than only a local region as in the previous section. A standard way to proceed is to detect local features, describe them and aggregate them. Existing approaches differ on how they perform each step and whether they are hand-crafted or learned. The rest of this section describes some applications and reviews the main standard descriptors. See [211] for an extensive literature review.

Figure 2.10: Example of day/night challenge for scene recognition. *Question*: which images match with the middle image: the left one or the right one? *Answer*: The left and middle image show the same Christmas tree. The right image shows another tree. A global descriptor defines a vector that summarizes the whole image. It is 'good' when the descriptors are similar for the first two images and different than the third one.

Applications. The main application for global features is scene recognition. It is the process by which a place that has been observed before can be identified when revisited. It is usually framed as an image retrieval problem: images from the visited scenes are

characterized by descriptors and stored in a database. When presented with a query image, the system retrieves the database images with the descriptors most similar to the query's one. The most popular example is the Google Image browser: you query Google with an image and it outputs the set of images depicting the same scene. Current applications work toward other types of outputs such as the Wikipedia page describing the scene content or a map location. In general, it serves as a pre-processing step before finer localization. In SLAM, it is used for loop closure, in SfM it helps to select the next image to register.

Hand-crafted Methods Early methods define global descriptors by aggregating local features such as SIFT. One of the first aggregation is inspired by text descriptions. For example, two texts are usually deemed similar if they exhibit similar word frequencies: a reinforcement learning paper is much likely to use the words 'state' and 'rewards' than a computer vision one. So a characteristic and compact representation of the text is a histogram of its text words. Then, two texts are similar if their histograms are. One way to choose the relevant words is to analyze word frequencies over a corpus and keep the most informative. An analog approach is adopted by the Bag of Words (BoW) [142, 172] approach, which is a gold standard for image retrieval. Here images are analog to text documents, and local features to words.

The first step is to compute the visual words over which the histogram is computed. Words are visual features computed by clustering the local features over a corpus of images. The second step is the image description. A new image is described by the distribution of its local features with respect to the visual words. For example, BoW computes a histogram over the visual words. Each feature is assigned to the nearest visual word and the bin corresponded to this word is increased. In practice, the image corpus used for this step is spatially disjoint from the image on which the scene recognition is performed.

The BoW approach has the advantage of being fast and relatively compact. For example, it is used for loop closure in ORBSLAM2 [131]. A typical value for the number of centroids, *i.e.* the descriptor size, is 64. This can still be too large when the number of images reaches several million. So a line of work is to reduce the dimensionality of such descriptor while preserving its discriminative properties [81, 143]. Another area for improvement is to refine the statistical model of the feature distribution over the visual words. Instead of a histogram, Fischer Vector (FV) [140, 141] fits a mixture of Gaussians over the visual words. The image descriptor concatenates the gradient of the probability

of the local features belonging to one of the Gaussians. Principal Component Analysis (PCA) is used to reduce the dimensionality of the vector.

Contemporary work introduces the Vector of Locally Aggregated Descriptors (VLAD) [82, 83] to improve the distribution model while maintaining the simplicity of BoW. As the authors put it, 'the VLAD is to the FV what k-means is to GMM clustering' [83]. The computation of the visual words stays the same. However, rather than only storing the cluster assignment as in BoW, they store the distance vector between each local feature and its nearest visual word. Authors show that this is a special case of the Fusher Vector derivations where all the Gaussians have the same weights and zero variance. The resulting descriptor is larger than in the previous methods so dimensionality reduction is necessary to make it practical.

In parallel to contributions on the feature distribution models, another line of work researches how to post-process the global descriptor. Reducing the dimensionality of the descriptor is one example. Another issue that arises is the strong correlation that can exist between local features. This occurs when, for example, the scene holds repetitive patterns, which leads to similar local features. The resulting descriptor is an unbalanced feature distribution over the words since similar features contribute to the same visual word. One solution is to whiten the features to reduce their correlation [79]. Other approaches rely on the descriptor normalization [10] or weight the feature contributions [188]. In addition to uniformizing the previous approaches under a common framework, [185] also investigates local descriptor selection schemes proposes to filter out the ones that do not contribute to the similarity measurement.

A common limit to existing approaches is their robustness to extreme variations in the image appearance such as day/night or seasonal changes. One reason is that the local features on which the descriptor relies are not robust to such variations either. One solution is to generate synthetic views of the query image to bring it closer to the database ones [187]. The next paragraph addresses another line of work that aims at improving local features. The contributions in Chap. 3 and 4 fall into this category. They break with the standard pixel-statistics based approaches and propose to leverage geometry and semantics to define local features.

Learning Methods Learning-based methods usually rely on the highly informative representation space of CNNs [57, 113].

One of the first contributions uses the CNN's fully connected layer's as the global image descriptor [16]. A first experiment shows that such a vector is as descriptive as

existing methods, even when the network is trained on an unrelated task, such as ImageNet classification [98]. A second experiment improves these descriptors by training the same network to classify landmark images over 1000 labels. The intuition is that such training constrains the fully connected layer to embed information relevant for the landmark recognition so it is better suited for retrieval. The resulting descriptor has a higher dimension that its predecessor but this is addressed with a simple PCA. The following work from the same authors then turns to manual aggregation of convolution features by sum-pooling rather than relying on the fully connected one [15]. Another way to aggregate these features is global max-pooling [14]. Although they reach satisfying performance, these aggregations fail to preserve the local feature locations, useful for geometric post-processing. This is addressed in [186] by computing image-region-specific descriptors: features are pooled over the region rather than the whole image. The global image descriptor is computed by the sum of all the regional features. This approach is refined by learning which regions are relevant to describe and fuse in the final descriptor [64]. One of the latest approaches [179] relies on a landmark detector to extract bounding boxes relevant to the scene recognition and select a subset of regions to keep with a filtering analog to ASMK [185]. Each region is described with a VLAD descriptor with the deep local features DELF [133] specifically trained for landmark image retrieval.

DELF follows the same philosophy but at the pixel-scale. It builds upon the previous landmark classification network [16] and fuses the convolutional features into an image descriptor with sum-pooling as in [15]. Their contribution resides in the training of importance weights for each feature before their aggregation. These weights represent how relevant a feature is for the landmark recognition. They are modeled with an additional layer that takes the last convolutional layer and outputs a weight for each feature. The weighted features are then fed to the same fully connected layer as before for output a landmark label. The network is trained on landmark classification: the authors fix the convolutional layers and only update the weights layers. This allows training the weights without direct supervision.

NetVLAD [9] relies on stronger supervision to train a deep version of VLAD [83]. Both local features and visual words are learned in an end-to-end fashion. VLAD computed local features over the images and accumulated the residues between each feature and its nearest visual word, computed in a previous step. In NetVLAD, dense local features are generated by a CNN and also assigned to visual words. However, the local features, the visual words, and the assignment are jointly learned. These operations must be differentiable to train the network and this is not the case for VLAD's hard assignment. So

it is replaced with a soft-assignment step and the final output is an image descriptor. The network is trained over image triplets where two images depict the same scene under various conditions and the third one is different. NetVLAD is optimized to generate similar descriptors for the matching images that are different from the third one by a margin manually defined. These image triplets are carefully selected at each step and require to know the camera poses for the training data. For a given query image, the positive image is the matching one with the nearest descriptor. And the negative image is the one with dissimilar content but with the nearest descriptor. The network is trained to bring the positive pair nearer and push the negative descriptor further.

Several works build upon NetVLAD. For example, [92] proposes to weight the NetVLAD feature before the soft-assignment step, which recalls the DELF philosophy. Another work aims at improving the delicate triplet selection, on which NetVLAD heavily depends on the triplet selection. For example, the camera poses available for most dataset have only GPS precision ($\sim$ 5 meters) so two images supposedly matching can have only a little or no overlap. This is addressed in [146] that leverages the 3D structure of the images: a pair of images depict the same scene when they depict the same 3D points. This allows modulating the 'positiveness' of the pairs, *i.e.*, how many points are common across images. The same applies to negative pairs and this selection is relevant to schedule the training on increasingly hard pairs.

Although the previous global descriptors reach impressive performance, it is complex to explicitly constrain the invariance of the local and global descriptors. This motivates following approaches to explicitly encode invariance properties in the descriptors using semantic information. In the same line as the selective descriptor approach, [129] weights the local descriptors depending on their semantic label before aggregating them in a BOW fashion. For example, in the context of urban scene recognition, vegetation features are down-weighted since they are more distracting than informative. [11] augments the features with semantic information to better filter obvious outliers. [182] uses semantics to mask the image and aggregate region descriptors. Regions are manually defined and the descriptor is made of a histogram over the pixel labels and a HoG-based descriptor over the masked pixels. VLASE leverages the pixel semantic distributions only to describe an image [201]: given that semantics stays mostly invariant to long-term appearance variations, the descriptor is relatively robust to such changes. They use a semantic edge network [202] to generate a distribution over each pixel. It represents the pixel probability to belong to a semantic edge of a given class. Local features are pixels with a probability higher than 0.5 to be an edge pixel and they are aggregated in a VLAD fashion.

Semantic Features for Localization Such semantic local features also benefit structure-based localization that are invariant over long periods and enable pixel correspondences that stay consistent over images with strong appearance variations. A second contribution of [182] is a structure-based camera pose optimization. Given a reference semantic 3D representation is available, it takes a query image with its initial pose computed with the retrieval procedure previously described. The pose is then refined by computing the camera pose that best projects the 3D semantic points to their corresponding pixels. A 3D point and a pixel correspond when they have similar labels. Although this reprojection error appears under-constrained, a careful selection of the labels to project provides satisfying localization results across seasons. These semantic correspondences are also used in [183] to make SfM more robust to appearance variations: the 2D-3D matches are scored by their semantic consistency. These values are used as weights during the RANdom SAmple Consensus (RANSAC)-based camera pose estimation: consistent matches are given higher importance. [162] embeds both semantic and 3D geometric information into a novel descriptor. They are generated by convolutional Variational-Auto-Encoder (VAE) trained for semantic volume completion. These local descriptors are then used in the standard SfM pipeline.

The semantic constraints can also be integrated into bayesian approaches. VSO [109] integrates them with the standard odometry ones to recover both the camera poses and the 3D representation over a set of images. The optimization loss has two terms: the first one embeds pixel matches based either on their intensity (direct approach) or their feature similarity (indirect). The second term constraints the camera poses and 3D locations to best approximate the semantic distribution over the 3D point clouds given dense semantic maps for each image. [175] follows the same approach and additionally takes advantage of the sequentiality of the images in a SLAM fashion. They integrate the semantic matches in a Bayesian filter where the observations are the pixel semantic labels. The 2D-3D semantic matches are leverages to define the observation probability. They show that this optimization achieves localization results on par with filters based on matches derived with standard high-dimensional features such as SIFT. Their filter has the advantage to rely on a much denser representation, which is pivotal for robotics applications. Following work [104] adapts the segmentation to make it better suited for this localization. In standard segmentation, the goal is to cut semantic units that have a human interpretation out. In the paper, a semantic unit is an arbitrary image region common to several images. For example, a tree segmented with the first approach would output a monolithic bloc over the tree. So any tree pixels can match with one another and this does not constrain the localization enough. With the second segmentation,

the trees would be divided into several semantic units: one for the trunk, one for the lower foliage and one for the upper foliage for example. This provides more matches constraints to exploit for the localization.

The two previous approaches are mostly applied to environments with a mix of urban and bucolic elements. This thesis differs in that it addresses the localization problem on bucolic scenes with few or no city structures. Also, it tackles the localization problem only at the image level, *i.e.*, the visual scene recognition problem. The pose of a query image is approximated with the one from the nearest database image. This approach was favored over the previous ones for it was deemed easier to tackle in these highly challenging images and helped to better grasp the visual properties of bucolic images. Research on finer localization is the object of future work.

Concurrent work [61] adopts a similar approach: it converts images into a semantic graph, uses temporal information to fuse the graphs over time and generates a global database graph. Then, given a new image expressed as a semantic graph, image retrieval is reduced to a graph matching problem. However, this approach assumes again that the environment is rich in semantic elements to avoid ambiguous graphs. This is not the case in bucolic environments which leads us to leverage edges as another robust and discriminative image signal.

2.4 Saliency

Visual modeling is the problem of learning how humans observe the world. One specific is saliency learning, *i.e.* learning the human's visual attention over pictures [29]. Attention is a general concept covering all factors that influence selection mechanisms, and saliency characterizes some part of a scene that stands out. Another line of work studies the gaze that is the coordinated movement of the eyes and the head. This section provides a brief review of saliency learning as it motivates the novel detector introduced in Chap. 5 of this thesis. Extensive reviews are available in [29, 59].

Saliency Learning There are mainly two categories of saliency: the bottom-up one and the top-down one. The bottom-up saliency represents the attractiveness potential of an image region. For example, the eye is automatically attracted to the soldier in the center of the image Fig. 2.11. One interpretation is that the peculiarity of such a figurine attracts our eye (bottom-up). Another explanation is that the figurine is the main information in the image and that the brain is trained to look at what is informative in the image (top-down).

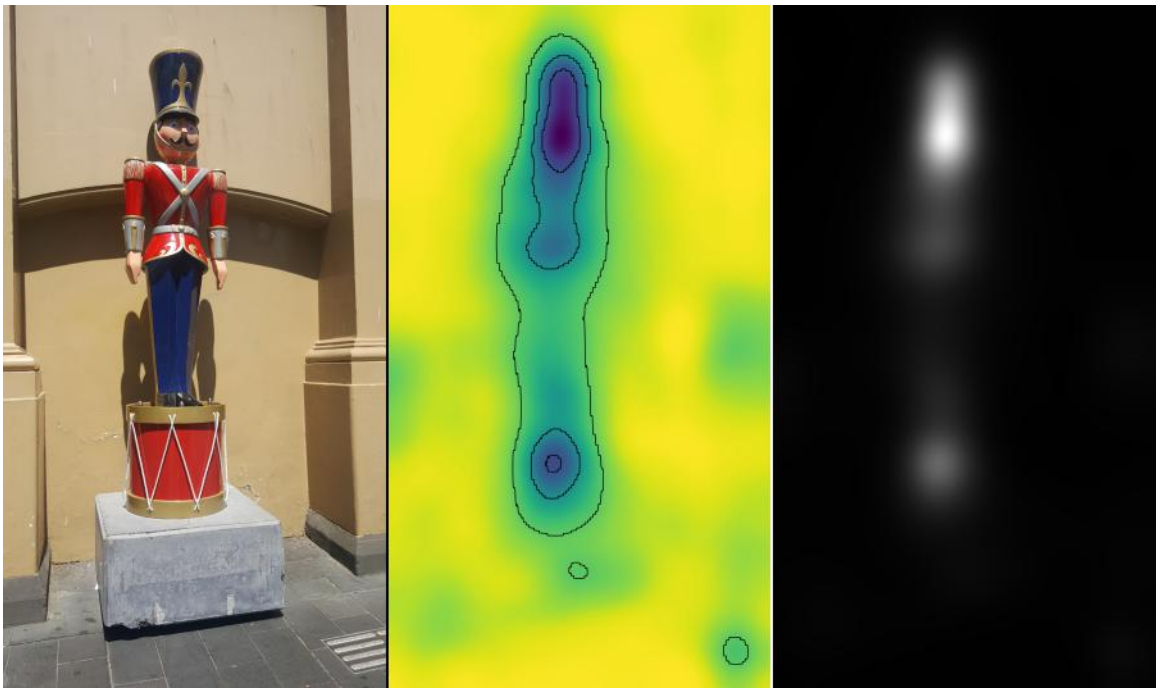


Figure 2.11: Left-Right: Soldier image - Color Saliency - Black-White Saliency. The saliency is generated using the DeepGazeII saliency model [101].

Early methods rely on low-level local features computed in a previous step such as pixel orientation, colors, and orientation to learn attention. Bayesian models [89, 208] learn the attention distribution over a set of images. It relies on a set of images with pseudo-ground-truth saliency collected either with manual labeling or with eye-trackers. Information-theoretic models maximize information sampled from the picture in an analog way to the information-driven exploration problem in robotics [97]. One application is autonomous visual exploration [45] where the next step is sampled in the most salient direction of the environment. Decision theoretic models define salient regions as the most useful one to complete a task [67].

The limit common to these approaches is that they rely on hand-crafted visual features to learn saliency. As for other vision domains, recent approaches propose to learn jointly the visual features and the saliency. DeepFix [99] and DeepGazeII [101] are among the best saliency models as ranked by the MIT Saliency Benchmark [87]. Despite recent efforts in uniformizing saliency metrics [31], evaluating such models is not as straightforward as camera pose regression error for example. One reason is that saliency is a more complex notion and each metric evaluates a property of the learned model.

A recent and popular application of saliency is the integration of 'attention models' in the training of unrelated models. DELF [133] is such an example where weights are

trained to give more importance to features relevant to the landmark's classification. This is analog to learning which features are the most salient where 'saliency' means 'visual usefulness'.

CNN Saliency Another popular application for saliency is CNN's visual interpretability. For example, it allows locating the image regions that motivate a trained classifier to say whether the image holds a cat or a dog. Although there is no quantifying metric for interpretability, it provides insightful qualitative results that can be used to pre-process images. Image regions deemed to hold useless information, or to be visual distractor can be masked. One example is object retrieval in cluttered environments [167]: given a cluttered scene, a trained CNN filters out most of the noise to keep information on the most salient objects. The saliency queried from the CNN can be used to mask the image before feeding it to a standard object retrieval pipeline.

Various computations have been explored to query trained CNN for saliency. The underlying idea is to compute the correlation between the CNN features and the image space, or between features at different levels. The approaches differ by their mathematical derivations and the handling of the non-invertible operations such as Rectifying Linear Unit (Relu) or max-pool.

One of the first visualizations aims at reconstruction images from logit scores and classification saliency [169]. The class saliency visualization queries the network about the spatial support of a particular class in the image. It is computed as the derivative of the class logits with respect to (*w.r.t*) the image space. This gradient has the same dimensions as the image. For each pixel, the gradient value expresses how much the class logit is correlated to this pixel, *i.e.*, how much this pixel contributes to the logit.

Assume that a trained classification network is fed with a cat image and classifies it as a cat. The last fully connected layer of the network generates a logit vector, where each entry is the probability for the image to belong to a class. In this example, the logits for the cat should be high and the others low. The authors aim at generating an image from scratch that will produce a logit with the same distribution. To do so, they learn the image pixels while freezing the network's weights. The image is initialized with zeros, fed to the network and iteratively updated to increase the cat logits. The pixel ranges are constrained with regularization. The optimization relies on gradient descent to update the pixels in the direction that increase the logits. This requires the computation of the derivative of the logit score *w.r.t* the image variables. This is achieved by backpropagating the logit score until the image space.

The following work proposes to visualize what contributes the most to feature activations instead of the classification result [206]. A trained classification network is fed with an image and generates a set of features maps and a classification score. The authors derive a computation to observe which of the image regions and structures 'excite' each layer of a given feature map. For each feature map, one layer is sampled and all the others are set to 0. The authors feed this modified feature map to a deconvolution network which can be seen as the network running the approximate inverse operation of the first classification network. They define alternative inverse for the non-invertible operations such as Relu and max-pool. In practice, the operation is similar to backpropagating the modified feature map *w.r.t* the image and where the backpropagation input is the modified feature map. The differences lie in the handling of the non-invertible operations. The result has the same dimension of the input image and with non-zero values on the pixels that contribute to this feature map. The derivation for ELF in Chap. 5 differs in that the whole feature map is backpropagated back to the image space without modification. Also, the output gradient is post-processed to make it better suited for local feature detection.

[117] builds upon the two previous works and reconstructs images from feature maps. It differs from [169] in that the image must generate a target feature map instead of target logits. It builds on the derivations [206] to compute the optimization gradients. Once again, an image is fed to a trained classification network to generate feature maps. The authors derive an image from scratch so that it leads to the same feature maps. They feed the trainable image to the network, compute a regression loss between the output features and the target ones, and backpropagate this loss through the network to the image space. The backpropagation outputs a gradient with the shape as the image. The image is iteratively updated until convergence and the resulting image has similar content as the target image. Similar optimization is employed in the neural style transfer [60] discussed in Chap. 7. Once again, ELF differs in that the feature map itself is backpropagated back to the image space and not a feature loss.

Instead of image reconstruction, the following works mostly focus on the gradient derivation for classification. [174] dubs its derivation 'Guided Backprop' and proposes to replace the pooling operations with large convolution stride to avoid the non-invertible pool operation. When backpropagating through Relu operations, they set the local backprop output to zero when the forward and the backward inputs are negative. Contrary to [206] that computes the derivative of the classification score of a label *w.r.t* the image, Grad-CAM [163] derives it *w.r.t* feature maps, and sums them with importance weights representing how much each map contributes to the classification score.

DeepLIFT [165] varies in that it defines reference values for the logits and the CNN feature maps, as when one sets the zero value of a balance. For example, a reference value can be the logits and the features generated from a noise image. When the trained network is fed with the image to classify, DeepLIFT computes the gradient of the difference between the logits and the reference ones *w.r.t* to the features difference. The authors argue that this provides a smoother gradient. Integrated gradients extend this derivation by accumulating the feature gradient *w.r.t* several intermediate inputs interpolated between the reference image and the image to classify. In practice, only a finite number of images are considered. One advantage is that the resulting gradient is less sensitive to image variations such as illumination changes. Similarly, Smooth-grad [173] averages several saliency maps over multiple instances of the same image disrupted with Gaussian noise.

All these approaches have in common that they derive saliency maps to explain a CNN's output. The various operations all aim for a smooth and fine delimitation of the image region that contributes the most to the CNN. Instead, this thesis leverages CNN saliency to define a novel local feature detector where features are located at the saliency's local maxima. It differs from the previous approaches in that it looks for salient regions independently of the image classification label. In ELF (Chap. 5), saliency is derived as the gradient of the feature map norm *w.r.t* the image. Previous methods usually compute the gradient of the classification score with either the feature maps or the image. The derivation most similar to ELF is the Guided Backprop [174]. A minimal difference lies in the Relu handling: ELF computes a simple backpropagation of the feature over the image space whereas the Guided Backprop masks the input and output of each local backpropagation based on their sign. The main difference with all the previous approaches is the investigation of the feature space of which saliency is suitable for local feature detection, and the post-processing of such a saliency map. The resulting local detector proves to be as relevant as when the CNN is specifically trained for feature detection.

Part I

Global Features Robust to Seasons

The first part of this thesis addresses the problem of bucolic scene recognition across seasons. It frames the problem as an image retrieval task where a query image under a specific light and season conditions is matched to the most similar image available in a database. This relies on the definition of a compact and informative summary of the image called a descriptor. Current retrieval methods are mostly tailored for urban environments for which the main challenge is to describe the image in a way that is robust to day/night variations and extreme viewpoints changes [84, 133]. Instead, the next two chapters focus on bucolics environments such as natural scenes with low texture and little semantic content. The nature of the variations is different and this leads to a different approach to image description. This part introduces two global image descriptors computed from the image's semantic and topological information. They achieve results on par with similar approaches [182, 201] on two multi-season datasets [66, 157] and even generalize to urban environments.

Chapter 3

Coarse Features for Long-Term Visual Scene Recognition[1]

Scene recognition is the process by which a place that has been observed before can be identified when revisited under different conditions. Visual features are used to describe images for place recognition. This task is usually cast as image retrieval where a query image is matched to the most similar image available in a database. The search is computed on a relevant image representation of much lower-dimensional space. Such encoding is usually the result of aggregating local image features whether hand-crafted or learned. The challenge is then to define features such that images of the same scene are near to each other despite their change of appearance due to environmental changes.

Most of the research effort on image-based place recognition is designed for urban environments [9, 16, 61, 82, 83, 133, 141, 172]. For these scenes, the main challenge is to design a global image description invariant to wide viewpoint variations and extreme illumination changes. However, robustness to seasonal variations is not critical because the image content is usually invariant to seasons. For example, a building is less likely to change over the course of the year than a tree. This is not the case for bucolic environments, such as natural scenes with low texture and little semantic content [66, 157]. There, the main challenge is to handle the variations in visual appearance across time such as illumination, weather, vegetation state in addition to the viewpoint changes. This chapter answers this problem by fusing hand-design and machine learning to define a global image descriptor based on semantics and geometry. The proposed approach selects semantic edges as relevant locations and describes them with their wavelet transform. This descriptor allows SoA performance for image retrieval on two multi-season environment-monitoring datasets: the CMU-Seasons and the Symphony Lake dataset.

[1] This chapter describes contributions to be published in ICRA 2020 [23].

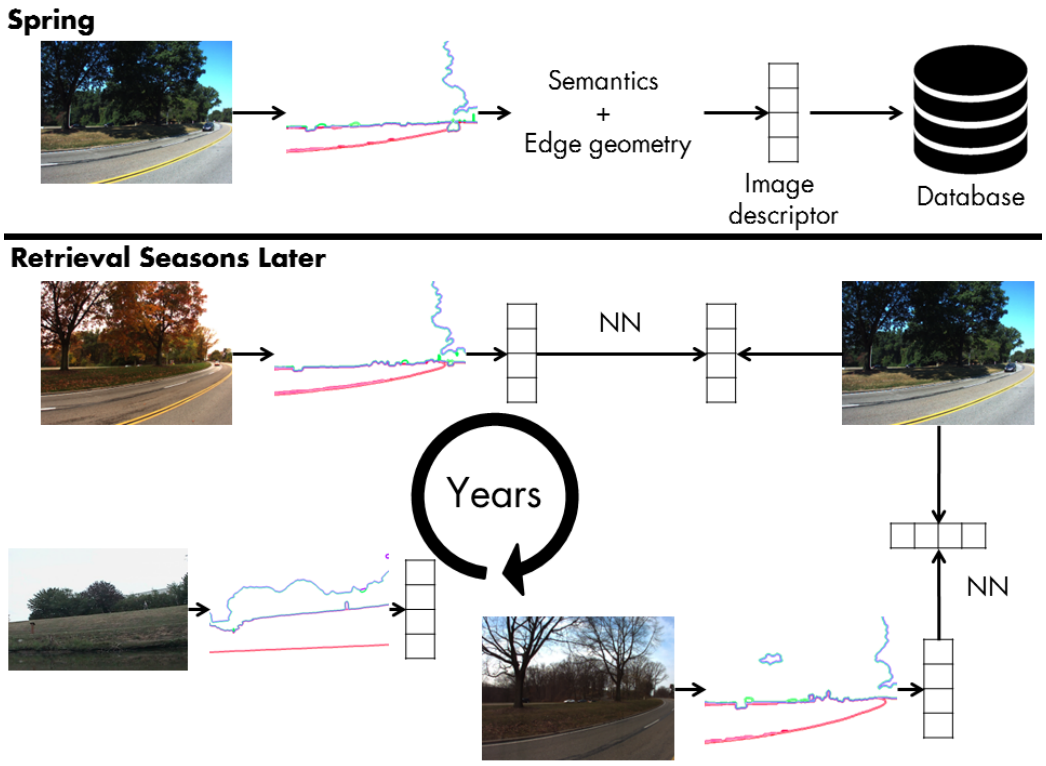


Figure 3.1: WASABI computes a global image descriptor for bucolic scene recognition across seasons. It builds upon the image semantics and its edge geometry that are robust to strong appearance variations caused by illumination and seasonal changes. While existing methods are tailored for urban-like scenes, our approach applies to bucolic scenes, which offer distinct challenges, and generalizes to city ones.

The usual approach for hand-crafted descriptors is to first detect and describe local features, then aggregate them into a low-dimensional vector. The methods differ in the local feature detection, description, and aggregation. Here, local features are the wavelet transform of the semantic edges. These edges are computed from the segmentation provided by a trained CNN. They are described by the wavelet transform [41] over a fixed-sized subsampling of the edge. This approach is motivated by the observation that edges and semantics are one of the most invariant information over long periods. So, it can be expected that these features are also robust to long-term variation in visual appearance. The global image representation is a simple concatenation of these edge descriptors and their labels.

Figure 3.1 illustrates the image retrieval pipeline with this novel descriptor dubbed WASABI[2] [3]. A collection of images is recorded along a road during the Spring. A de-

[2]WAvelet SemAntic edge descriptor for BucolIc environment
[3]https://github.com/abenbihi/wasabi

scriptor is computed for each image and stored in a database. Later in the year, *e.g.* in Autumn, while traversing the same road, the image at the current location is described. The goal is to retrieve the database image which descriptor is the nearest to the current one. The image distance is computed by first assigning each semantic edge from the first image to the nearest one in the second image based on their wavelet descriptor distance. An additional constraint is that associated edges should have the same semantic label. The distance between the two edges is the Euclidean distance between their descriptors. Finally, the image distance is the sum of the distances between edge descriptors of associated edges. Note that this computation can seem heavy and is one of the limits addressed by the next descriptor in Chap. 4.

The rest of this chapter is organized as follows. Sec. 3.1 recalls the subset of SoA described in Chap. 2 that is relevant to this chapter. Then Sec. 3.2 details the visual features derivation. In Sec. 3.3, WASABI is compared to existing image retrieval methods on two outdoor bucolic datasets: the park slices of the CMU-Seasons [157] and Symphony [66], recorded over a period of 1 year and 3 years respectively. Experiments show that it outperforms existing methods, both hand-crafted and learned even when the latter are finetuned for these datasets. It is also on par with NetVLAD, one of the current SoA, on urban scenes, which is specifically optimized for city environments. This shows that WASABI can also generalize across environments.

3.1 Review of visual scene recognition

This section summarizes the place recognition SoA previously detailed in Sec. 2.3. All image retrieval methods follow roughly the same steps: local feature detection and description, and feature aggregation into a global image descriptor. They differ in how they perform each step and whether it is hand-crafted or learned.

Hand-crafted approaches. Early global descriptors are designed by aggregating locally invariant features such as SIFT [116]. The first step is the generation of the visual words by clustering local feature descriptors over a training dataset. The words are the clusters' centroids and are usually referred to as the codebook. The training dataset must be spatially disjoint from the place recognition one to generalize well. An image is then described with the statistics of its local features with respect to this codebook. In BoW [172], the local features of the image are assigned to the codebook clusters and the descriptor is simply the histogram of cluster occurrences. The Fisher vectors [141] improve over the previous clustering by fitting a mixture of Gaussians over the visual words.

Then, for each local feature of the image of interest, they concatenate the gradient of the probability of this feature to belong to one of the Gaussians. This high-dimensional vector is then reduced with Principal Component Analysis (PCA). This approach is simplified in VLAD [82] that concatenates the distance vector between each local feature and its nearest cluster. All these methods rely on features based on pixel distribution that assumes that images have strong textures, which is not the case for bucolic images. They are also sensitive to variations in the image appearance such as seasonal changes. In contrast, WASABI relies on the image's local geometry and semantics, which proves to be robust to strong appearance changes.

Learned approaches. Later works leverage the rich representation space of CNN to design higher-level local features. This aims at disentangling local features and pixel intensity through learned feature descriptions. [16] uses the features of a pre-trained classification CNN as local features and aggregates in a VLAD fashion. VLASE [201] follows the same approach and aggregates local features extracted from the CaseNet network [202], trained to generate semantic edges. Keypoints are pixels that lie on a semantic edge and they are described with the probability distribution of the pixel to belong to a semantic class, as provided by the last layer of the CNN. DELF [133] trains the network to classify landmarks with a weight layer to give more importance to informative CNN features. Those with high weights are then aggregated as in VLAD. NetVLAD [9] proposes to train both the CNN features, the visual words, and the aggregation.

They define an end-to-end learning pipeline and reach top performances on urban scenes such as the Pittsburg or the Tokyo time machine datasets [187, 188]. WASABI also relies on CNNs but only to segment images, not to describe them. In this sense, it is similar to the image description defined in [182]: the image is divided into patches over which one semantic descriptor and one pixel-statistics-based descriptor are computed. The first one is the semantic histogram of the static semantic classes, and the second one is the HoG of the patch, as in SIFT. The global image descriptor is the concatenation of the patch descriptors. WASABI differs in that it describes the geometric properties of the semantic edges and not pixel statistics. The edge wavelet transforms are accumulated to describe the image.

Edge description. The choice to represent edges with their Wavelet transform is motivated by experimenting with various edge descriptors. Most of them, reviewed in [123], are tailored for simulation-like images where edges are smooth rather than outdoor images where they are much noisier. This explains why edge descriptors are usually less

robust to illumination and viewpoint variations than their pixel-based counterparts. Experiments show that the wavelet descriptor [41] is invariant and compact enough to describe an edge in its entirety and be integrated into the image retrieval pipeline.

Chap. 4 will show that local edge information is even more relevant to integrate in the image descriptor. Inspired by contributions on loop closing using lidar local descriptors [88, 181, 190], the next chapter defines local edge features described with the Shape Context Descriptor [22]. Whether in this chapter or the next, the influence of the edge description on the retrieval performance is still under investigation.

3.2 Scene Recognition from Coarse Hand-Crafted Features

This section details the derivation of WASABI from visual features robust to long-term variations in appearance. Semantic edges are extracted and described by their wavelet transform. An image is characterized by this set of edge descriptors and their semantic labels. Two images are similar when the distance between matching semantic edge descriptors is small.

3.2.1 Semantic Edges as Regions of Interest

Given a color image as input, local features are the continuous semantic edges described with the wavelet transform. To extract the edges, two equivalent approaches can be considered. The first is to extract them from the semantic segmentation of the image, i.e. its pixel-wise classification [103, 210]. The second approach is also based on CNNs but directly outputs the edges together with their labels [6, 202, 203]. The first approach is favored for the following reasons. First, there are many more public segmentation models than semantic edge ones. This generalizes WASABI to a wider range of data by substituting the segmentation model with one tuned to the data. Also, semantic edges generated with the second approach are coarse and noisy so they are less consistent and repeatable than the ones from the first method.

Starting from the semantic segmentation, a post-processing stage is necessary to reduce the labeling noise. Most of this noise consists of labeling errors around edges or small holes inside bigger semantic units. To reduce the influence of these errors, semantic blobs smaller than `min_blob_size` are merged with their nearest neighbors.

Furthermore, to make semantic edges robust over long periods, it is necessary to ignore classes corresponding to dynamic objects such as cars or pedestrians. Otherwise, they would alter the semantic edges and modify the global image descriptor. These

classes are removed from the segmentation maps and the resulting hole is filled with the nearest semantic labels.

A simple Canny-based edge detection is run on the cleaned-up semantic segmentation and edges smaller than `min_edge_size` pixels are filtered out. Segmentation noise may also break continuous edges, so the remaining edges are processed to reconnect edges belonging to each other. For each class, if two edge extremities are below a pixel distance `min_neighbour_gap`, the corresponding edges are grouped into a unique edge.

The parameters are chosen empirically based on the segmentation noise of the images. Images are segmented with the PSP-Net [210] model trained on cityscapes [43] and finetuned for robust inter-season segmentation [103]. In this case, the relevant detection parameters are summarized in Table (Tab.) 3.1.

`min_blob_size`	50
`min_edge_size`	50
`min_neihbor_gap`	5

Table 3.1: Edge sampling parameters.

3.2.2 Feature Description: Edge Wavelet Transform

Among the many existing edge descriptors, the wavelet descriptor [41] is favored for its properties relevant to image retrieval. It consists of projecting a signal over a basis of known functions and is often used to generate a compact and unique representation of a signal. Wavelet description is not the only transform to generate a unique representation for a signal. The Fourier descriptors [65, 205] also provide such a unique embedding. However, the wavelet description is more compact than the Fourier one due to its multiple-scale decomposition. Experiments confirmed that the former was more discriminative than the latter for the same number of coefficients.

The 2D contours extracted from the semantic segmentation are subsampled at regular steps to collect N pixels. Their (x, y) locations in the image are concatenated into a 2D vector. The discrete Haar-wavelet decomposition is computed over each axis separately. The two output vectors are concatenated and L2 normalized. In the experiments, N = 64 is used and only the even coefficients of the wavelet transforms are kept. This does not destroy information as the coefficients are redundant. The final edge descriptor is a 128-dimension vector.

3.2.3 Feature Aggregation and Image Distance

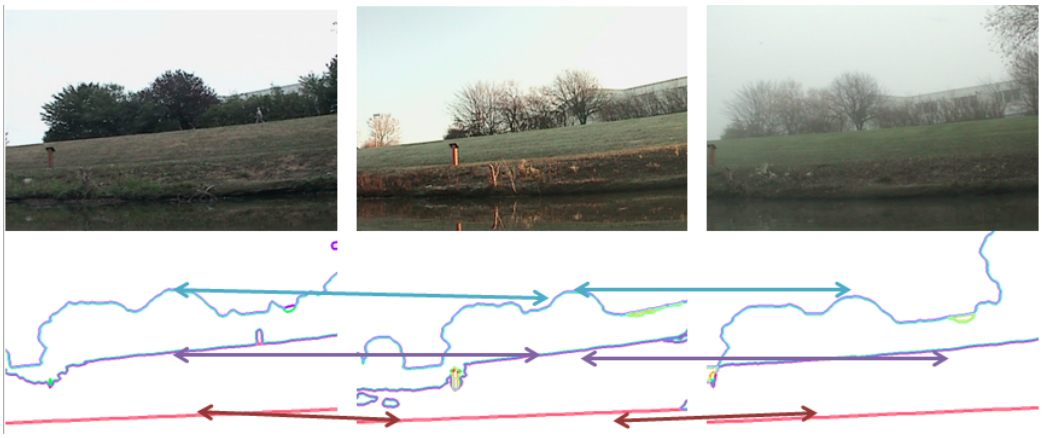


Figure 3.2: Symphony. Semantic edge association across strong seasonal and weather variations.

The image descriptor is the accumulation of the edge's wavelet transforms and semantic label. Given two images and their aggregated edge descriptors, the image distance is the average distance between matching edges. Matching edges are computed by solving the assignment problem between edges of the same class (see Fig. 3.2). The distance used is the Euclidean distance between edge descriptors and the image distance is the average of the associated descriptor distances. In a retrieval setting, we compute such a distance between the query image and every image in the database and return the database entry with the lowest distance.

3.3 Experiments

WASABI shows better performance on bucolic scenes than existing methods while only slightly underperforming NetVLAD and DELF on urban environments. This is expected as they are optimized for such settings. Still, this shows that fusing hand-design and machine learning can provide visual features as relevant and robust as the most performing learning approaches. Finetuning the learning methods on the bucolic scenes proves to be useful for VLAD but does not improve the overall performance for BoW and NetVLAD. A plausible explanation is that these methods require more data than the one available. Note however that, in addition to robustness, this highlights another advantage of integrating hand-design in visual features definition: it reduces the burden of data collection and training. The rest of this section presents the numerical results and their interpretation.

Datasets. This paragraph describes the two bucolic datasets over which the image retrieval is evaluated.

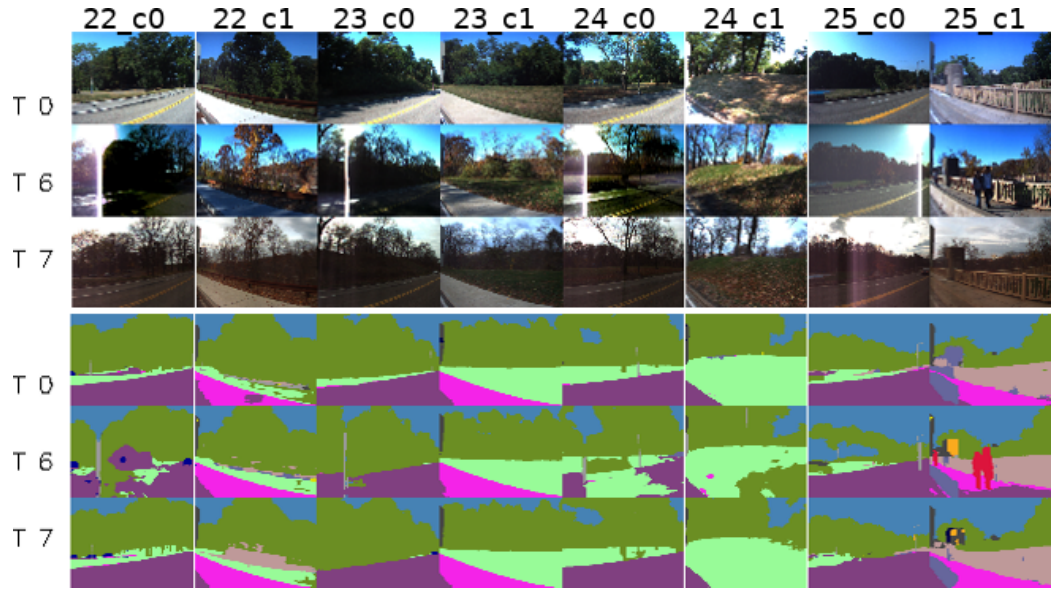


Figure 3.3: Extended CMU-Seasons. Top: images. Down: segmentation instead of the semantic edge for better visualization. Each column depicts one location from a slice `i` and a camera `j` that we note `i_cj`. Each line depicts the same location over several traversals noted `T`.

Extended CMU-Seasons. The Extended CMU-Seasons [157] dataset (Fig. 3.3) is an extended version of the CMU-Seasons [17] dataset. It depicts urban, suburban, and park scenes in the area of Pittsburgh, USA. Two front-facing cameras are mounted on a car pointing to the left/right of the vehicle at approximately 45 degrees. Twelve traversals are recorded over a period of 1 year and the images from the two cameras do not overlap. The traversals are divided into 24 spatially disjoint slices, with slices {2-8} for urban scenes, {9-17} for suburban and {18-25} for park scenes respectively. All retrieval methods are evaluated on the park scenes for which ground-truth poses are available {22-25}. The other park scenes {18-21} can be used to train learning approaches. The reference traversal is recorded during a sunny day in early spring with no foliage on the trees. The 11 other traversals are the queries and cover all seasons with various illuminations and light artifacts summarised in Table 3.2. Note that the 11-th traversal captured from the right-camera is much shorter than the 10 others (10 images vs 200 images), so we discard it in the evaluation. In total, there are 80 image sets of roughly 200 images with ground-truth camera poses. Figure 3.3 shows examples of matching images over multiple seasons with significant variations in season and lighting.

Traversal	Season	Tree State	Weather / Light	Artifact
Reference	Early Spring	No Foliage	Sun	-
0	Spring	Foliage	Sun	-
1	Early Autumn	Foliage	Overcast	-
2	Autumn	Foliage	Sun	Light haze
3	Autumn	Mixed-Foliage	Sun	-
4	Autumn	Mixed-Foliage	Overcast	-
5	Autumn	Mixed-Foliage	Low-Sun	-
6	Autumn	Mixed-Foliage	Sun	Sun glare
7	Winter	No Foliage	Overcast	Light haze
8	Winter	No Foliage	Snow / Sun	Sun glare
9	Winter	No Foliage	Sun	-
10	Spring	Foliage	Overcast	Light haze

Table 3.2: CMU-Seasons season and light conditions for each traversal.

Symphony. The Symphony [66] dataset consists of 121 visual traversals of the shore of Symphony Lake in Metz, France. The 1.3 km long shore is surveyed using a pan-tilt-zoom (PTZ) camera and a 2D LiDAR mounted on an unmanned surface vehicle. The camera faces starboard as the boat moves along the shore while maintaining a constant distance. The boat was deployed on average every 10 days from Jan 6, 2014 to April 3, 2017. In comparison to the roadway datasets, it holds a wider range of illumination and seasonal variations and much less texture and semantic features, which challenges existing place recognition methods.

For this evaluation, 10 discontinuous traversals are randomly sampled over the East side of the lake using the ground-truth poses computed in [145]. The West side of the lake can be used for training. To define the database, images from one of the 121 traversals are sampled at a regular interval. For each database image, the matching images are sampled from 10 random traversals out of the 120 left. Note that contrary to the CMU-Seasons dataset, this means that there is no light and appearance continuity over one traversal (Fig. 3.4).

The experiments evaluate WASABI against the SoA over a wide range of season and illumination variations on the Extended-CMU-Seasons and the Symphony datasets [157, 66]. The CMU-Seasons dataset even allows for a finer evaluation of the performances with respect to semantic in one hand, and season and illumination on the other hand.

Baselines. WASABI is compared to SoA image retrieval methods BoW, VLAD, NetVLAD and DELF [9, 83, 133, 172]. In their version available online, these methods are mostly

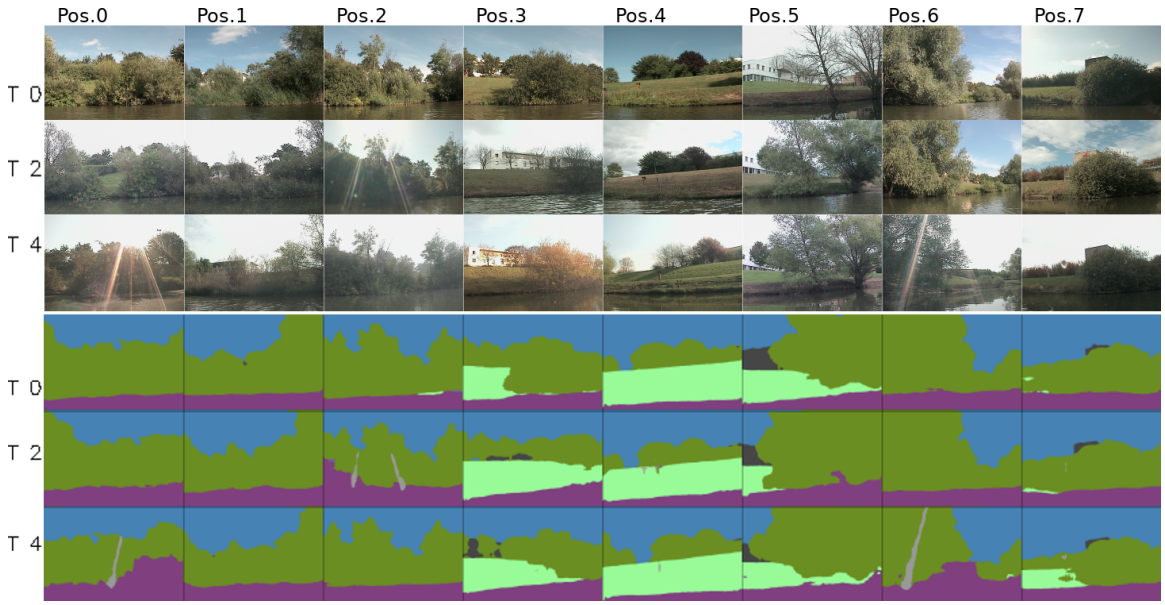


Figure 3.4: Symphony dataset. Top-Down: images and their segmentation. First line: reference traversal at several locations. Each column `k` depicts one location `Pos.k`. Each line depicts `Pos.k` over random traversals noted `T`. Note that contrary to CMU-Seasons, we generate mixed-conditions evaluation traversals from the actual lake traversals. So there is no constant illumination or seasonal condition over one query traversal `T`.

tailored for rich semantic environments: the codebook for BoW and VLAD is trained on Flickr60k [80], NetVLAD is trained on the Pittsburg dataset [188] and DELF on the Google landmark one [133]. For fair comparison, we finetune them on CMU-Seasons and Symphony when possible, and report both original scores and the finetuned ones noted with `'_tuned'`.

BoW and VLAD. A new codebook is generated for BoW [172] and VLAD [83], by clustering SIFT local features on 1691 images from the few CMU park training slices {18-21} and 1230 images from the West side of the Symphony lake. The number of clusters is set to 64.

NetVLAD. This paragraph only summarises how NetVLAD is finetuned on the CMU-Seasons and Symphony datasets. A detailed and self-contained description of the finetuning of NetVLAD is available in Appendix A. These details are not necessary for the rest of this chapter but are included for the sake of reproducibility.

NetVLAD is trained to generate a global description from an input image. It is optimized with the triplet loss [20]: a set of three images is sampled from the dataset so that

two of them match and the third one depicts another scene. The network is optimized so that the descriptors of the matching images are near to each other while the third one is far. Finetuning requires images with ground-truth poses, which is only the case for slices {22-25} of the CMU-Seasons. So the model is trained on three slices from {22-25} and evaluated on the remaining one. For each configuration, half of the images are used for the database and the other half as queries. On Symphony, images together with their ground-truth poses are sampled from the west side of the lake that is spatially disjoint from the evaluation traversals.

DELF. The DELF learned local features are not finetuned because the training code is not available and we did not manage to reproduce it when replacing the landmark scenes to classify [16] with scenes sampled from the CMU-Seasons and Symphony traversals. One probable explanation is that these datasets are too small to ease the convergence.

The authors provide four codebooks: two made of 1024 words, each trained either on the Paris6k dataset or the Oxford5k dataset, and two others made of 66536 words. In both cases, the visual words have dimensions 128. The raw DELF features have dimension 2048 so they are quantized into a vector of dimension 128 using the dimensionality reduction code provided by the authors. Only the two 1024-codebooks are tested and they lead to near-equal quantitative performance so we report only result for the Paris6k codebook.

VLASE. VLASE extracts local features from the CaseNet CNN [202] trained to generate semantic edges over an image. VLASE aggregates these features in a VLAD fashion.

Given an image $\mathrm{I} \in \mathbb{N}^{h \times w \times 3}$ of height $h \in \mathbb{N}$ and width $w \in \mathbb{N}$, K semantic labels, the network outputs K pseudo-probability maps $(\mathrm{Y}_k)_{k \in [\![0, \mathrm{K}-1]\!]}$, with $\mathrm{Y}_k \in [-1,1]^{h \times w}$. Each map Y_k represents the probability of the pixels to belong to a semantic class k, *i.e.*, $\mathrm{Y}_k(u,v)$ is the probability that the pixel (u,v) belongs to the semantic edge of class k. Note that the CaseNet network is designed such that $\sum_{k=0}^{\mathrm{K}-1} \mathrm{Y}_k(u,v) \neq 1$, *i.e*, a pixel can belong to several classes with probabilities higher than 0.5 for each. Conversely, a pixel can belong to no edge at all when $\mathrm{Y}_k(u,v) = 0, \forall k \in [\![0, \mathrm{K}-1]\!]$. One possible motivation is that edge pixels lie at the limit between at least two semantic classes so it seems natural to assign these pixels more than one label. When the probabilities are constrained to sum to one, it may be harder to distribute the probability weights over several classes. Instead, optimizing the independent classification of a pixel on whether it belongs to a

semantic edge with label k seems more straightforward. The labels CaseNet is trained on are the Cityscape labels described in Tab. 3.3.

road	pole	terrain	truck	car
sidewalk	traffic light	sky	bus	fence
building	traffic sign	person	train	
wall	vegetation	rider	motorcycle	

Table 3.3: Cityscapes labels.

The authors define a local feature as a pixel for which there is at least one class c such that the probability to belong to a semantic edge is higher than a threshold T_e. More formally, a pixel (u,v) is a local feature if $\exists k \in [\![0, K-1]\!], Y_k(u,v) > T_e$. In practice, there are too many such pixels, in the order of $[10^4, 10^5]$, and many of them are strongly correlated, especially when they are contiguous. This motivates the choice to keep only up to `max_num_feat` such pixels.

A grid search over the two parameters T_e and `max_num_feat` are run over the ranges $T_e \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, `max_num_feat` $\in \{1000, 2000, 3000, 4000, 5000\}$. The number of words n_{words} in the visual codebook is also tested with $n_{words} \in \{32, 64\}$. The best performances are reached with the following set of parameters Tab. 3.4.

T_e	0.5
`max_num_feat`	3000
`n_words`	64

Table 3.4: VLASE parameters.

Toft *et al.* The localization paper [182] relies on image retrieval to initialize pose optimization. They define a global image descriptor from semantic and pixel intensity statistics. In the paper, the top half image is divided into 6 rectangle patches (2 lines and 3 columns). For each patch, they first compute a semantic histogram over the static classes, such as the vegetation or the road. Then they compute a descriptor based on the pixel statistics: the patch is masked to keep only the vegetation and building pixels and the histogram of oriented gradients of these pixels is computed, as in SIFT. More specifically, each patch is further divided into smaller rectangles over which a HoG is computed. The patch descriptor is the concatenation of the semantic histogram and all the HoGs. Then, the image descriptor is the concatenation of the patch descriptors.

This descriptor depends on the parameters listed in Tab. 3.5 for which a grid search in run.

Parameters	Grid search range	Optimal value	Authors' parameters
Number of patches per line	$[\![2,6]\!]$	6	2
es Number of patches per column	$[\![3,9]\!]$	9	3
(HoG) Number of rectangles per line	$[\![2,4]\!]$	4	4
(HoG) Number of rectangles per column	$[\![2,4]\!]$	4	4
(HoG) Bin discretization	8	8	8
Descriptor Length	-	7506	834

Table 3.5: Toft *et al.* parameters over image of size $h = 768$ and $w = 1024$. Note: the optimal values corresponds the highest resolution tested. It is expected that the quantitative performance would be even better with a higher resolution.

The description is computed over the top two-third of the image instead of the top half. This crop provides the best retrieval results over the crops tested by sliding horizontal windows of various sizes over the image. The semantic histogram is computed over all the static classes. The HoGs are computed over the pixels belonging to the vegetation and building classes. The addition of other static classes in the HoG does not improve the performance for this dataset.

Note that the best parameters are the ones that lead to local description at the highest resolution: this method performs best with the maximum number of patches and the maximum HoG discretization. A descriptor with higher resolution would probably give even better results but this would come at the cost of a longer descriptor: here, one image descriptor already has a dimension of $6 \cdot 9 \cdot (4 \cdot 4 \cdot 8 + 11) = 7506$. In practice, one would investigate quantization to reduce this descriptor dimension. This is outside the scope of this chapter and all descriptors are compared with their original dimensions. Another variant of this method is run with the author's parameters.

Metrics. The place recognition metrics are the *recall*@N and the mean Average Precision (*mAP*) [142]. Both depend on a distance threshold ϵ: a retrieved database image matches the query if the distance between their camera center is below ϵ. Both metrics are available in the code [4].

[4] https://github.com/abenbihi/wasabi

The *recall@*N is the percentage of queries for which there is at least one matching database image in the first N retrieved images. We set N $\in \{1,5,10,20\}$, and ϵ to 5 meters (m) and 2m for the CMU-Seasons and the Symphony datasets respectively.

The *mAP* frames image retrieval as a classification problem. Given a query image, let P be the total number of database images matching this query and N the number of retrieved images. Using classification syntax, true positives are the retrieved images that match the query. False positives are the remaining retrieved images. Note that N = TP + FP. With the standard definitions of precision and recall, the *mAP* is the area under the precision-recall curve. More formally, let TP and FP be the number of true and false positives respectively. The precision is the ratio $\frac{TP}{TP+FP}$ and the recall is $\frac{TP}{P}$. The precision-recall curve is drawn by computing these ratios for several values of retrieved images N. This *mAP* implementation is borrowed from the code of [80].

Setup. The segmentation is generated with the PSP-Net [210] specifically finetuned for segmentation robust to Seasons on the CMU-Seasons dataset provided by [103]. Given the lack of ground-truth segmentation on the Symphony dataset, the segmentation is generated using the same model for both datasets. Although the results on the Symphony dataset are noisier, WASABI still manages to provide SoA results.

The GPU-based approaches are run on an Nvidia 1080Ti with Torch 0.4.1 for the segmentation, Tensorflow 1.12, Cuda9 and Cudnn7 to train NetVLAD, and Tensorflow 1.5 with the V1.13 tag from the DELF GitHub repository. The CPU-based methods are run with Python3 and OpenCV 3.4..

3.3.1 Global evaluation on Extended-CMU-Seasons

Experiments suggest that a hand-design approaches based on semantics, such as the WASABI descriptor and [182, 201], are as relevant for scene recognition as deep-learning approaches [9, 133].

Fig. 3.5 plots the *Recall@*N over the three types of data: the CMU park, the Symphony lake, and the CMU city. Overall, the method from Toft *et al.* achieves the best results when it aggregates local descriptors at a high resolution (toft_etal (7506)). The necessary memory overhead may be addressed with dimensionality reduction but this is out of the scope of this chapter. WASABI achieves the 2^{nd} best performance of the CMU park and is on par with SoA methods tailored for urban environments such a NetVLAD and DELF. This suggests that semantic edges are discriminative enough to recognize a scene, even when there seem to be few semantic elements such as in the park. This assumption is comforted by the satisfying performance of VLASE, which also leverages semantic edges

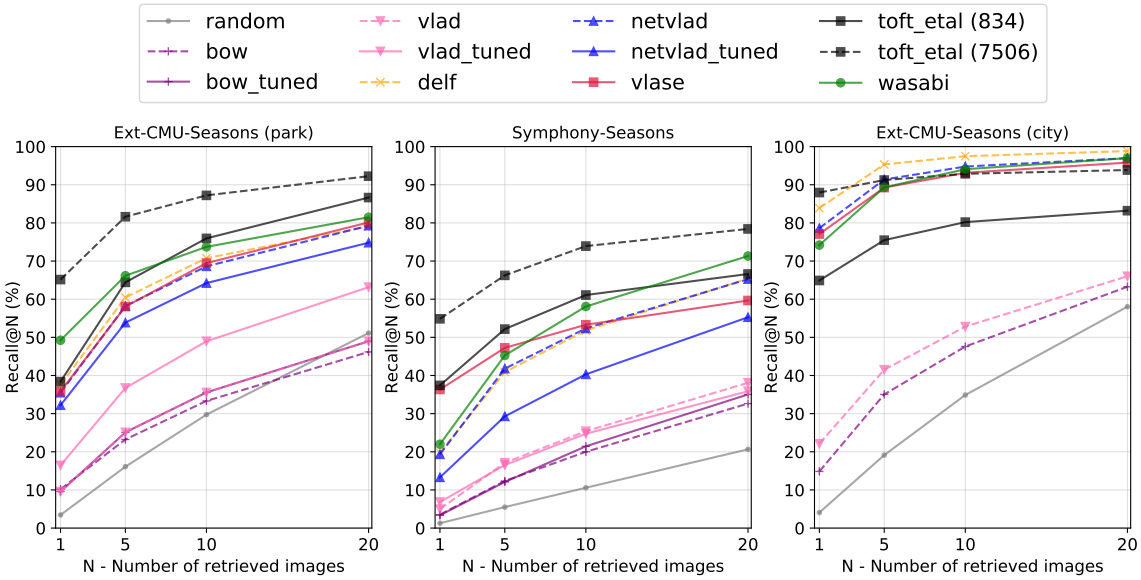


Figure 3.5: Retrieval performance for each dataset measured with the *Recall*@N. Retrieval is performed based on the similarity of the descriptors and no further post-processing is run for all methods. The high-resolution description from [182] reaches the best score, followed by WASABI and current SoA methods. These results suggest that a hand-designed descriptor can compare with existing deep approaches. However, WASABI still needs to be improved to be as relevant as Toft *et al.* [182]'s description.

to compute local features. Note that while it underperforms WASABI on the CMU-Park, it provides better results on the Symphony data and the same goes for [182].

There are two main explanations for the poor WASABI results on Symphony: the first is that the segmentation model trained for the CMU images generates noisy outputs on the Symphony images, especially around the edges (Fig. 3.6). So the WASABI wavelet descriptors cannot be consistent enough across images. One reason that allows [182] and VLASE to be robust to this noise is that they do not rely on the semantic edge geometry directly: Toft *et al.* leverages the semantic information in the form of a label histogram which is less sensitive to noise than segmentation itself. A similar could explain VLASE's robustness even tough it samples local features from those same semantic edges. The final histogram of semantic local feature is less sensitive to semantic noise than the semantic edge coordinates on which WASABI relies.

The second explanation for WASABI's underperformance on WASABI is the smaller edge densities compared to the CMU data. This suggests that the geometric information should be leveraged at a finer scale than the edge's one. This is addressed in the next chapter along with the scalability issues. Note that the descriptor size and the image distance computational complexity are quadratic with the number of semantic edges of

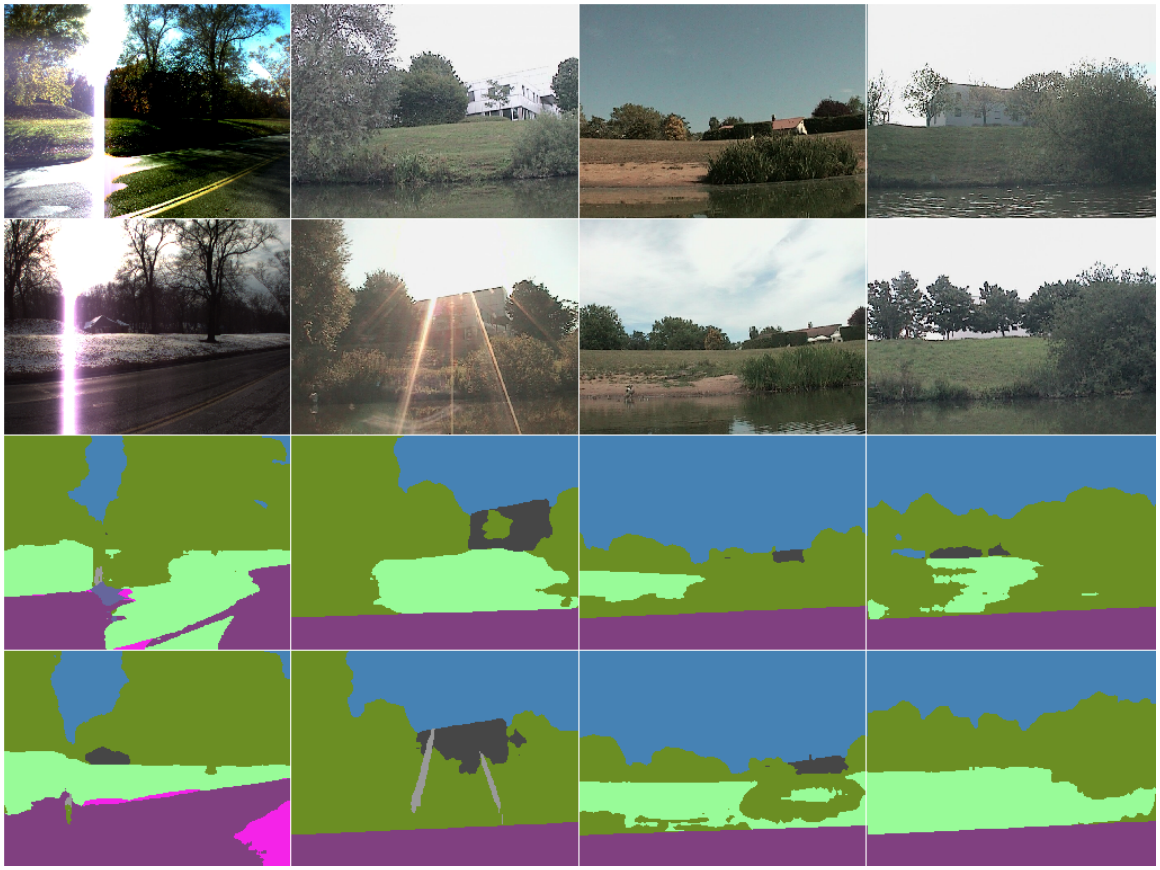


Figure 3.6: Segmentation failures. Left column: In Ext-CMU-Seasons, strong sun glare is present along traversals 6 (sunny spring) and 8 (snowy winter). Other columns: Symphony. The segmentation is not finetuned on the lake and produces a noisier output. It is also sensitive to sun glare.

the image. This prevents its integration in real systems and this problem is addressed in the Chap. 4. The rest of this section analyzes the influence of image variations on retrieval performance.

3.3.2 Robustness to Illumination Variations

Light / Season	Spring	Autumn	Winter
Overcast	10	2,4	7,9
Sun	0	1,3,5,6	8

Table 3.6: CMU-Seasons traversal ids and their season and light conditions.

Rationale. The main sources for visual variations in a bucolic scene are the light and the season. The Ext-CMU-Seasons allows evaluating the robustness of the visual features with respect to these two factors. Each car traversals captures scenes from urban and bucolic environments with a specific season and light condition (*c.f.* Tab. 3.2). The retrieval results on traversals with similar conditions are averaged. Table 3.6 recalls the traversals that share visual appearance.

Results. The next paragraph discusses the robustness of WASABI and SoA methods to illumination and season variations. Overall, WASABI is as robust as existing methods.

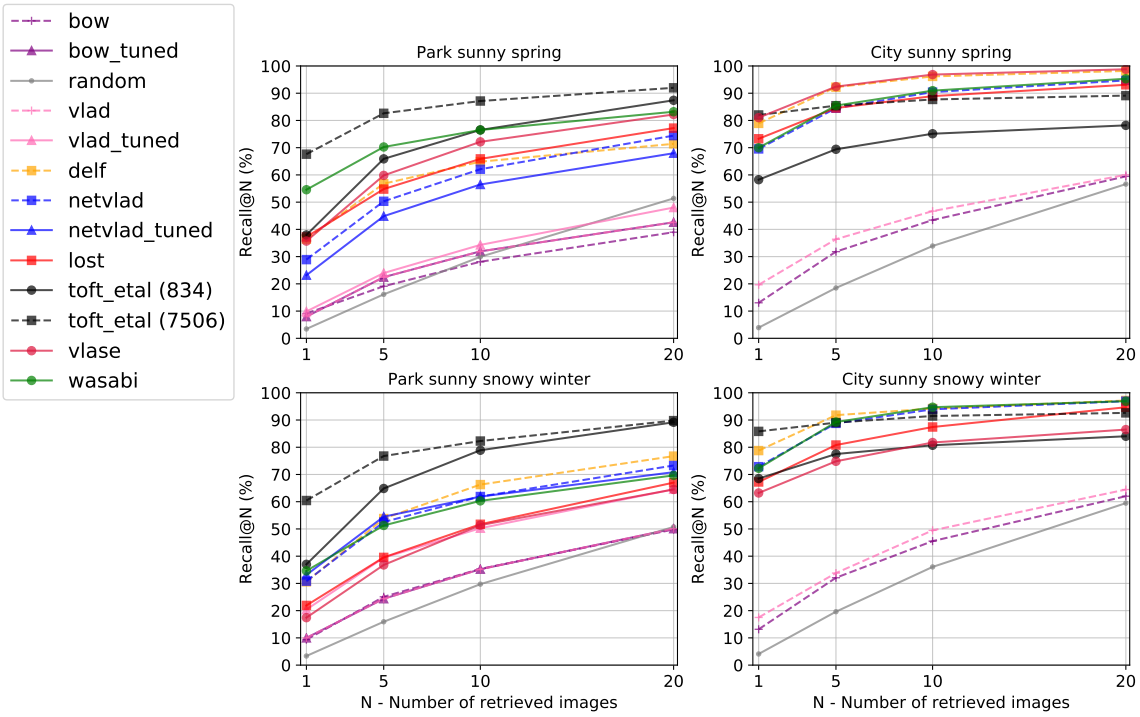


Figure 3.7: CMU-Seasons on Sunny Weather. The reference traversal is sampled during winter. The degradation performance in the Park during the winter snowy weather (bottom-left) is mostly due to the destructive sun glare in the images. The weather seems to hold no influence on city retrieval results.

Various seasons on Sunny Weather. Figure 3.7 suggests that all methods are more sensitive to light variations than seasons. The main difference between the spring traversals (top) and the winter ones (bottom) is that the winter images of the park suffer from strong sun glare in most of the images (Fig. 3.6 - left column). This is damageable for pixel intensity-based approaches like NetVLAD but also WASABI. This artifact degenerates the semantic edges so all the derivations following are corrupted. Note however

that the results are not impacted in the city. Although some city images exhibit the sun glare, it is much less present and the images hold enough semantic elements to recover from it.

Given that the reference traversal was sampled during winter, these results suggest that local light artifacts have a higher impact on the recognition performances than global scene variations like the leaves' color. This suggests that descriptors robust to general image noise is as important as robustness to appearance variations.

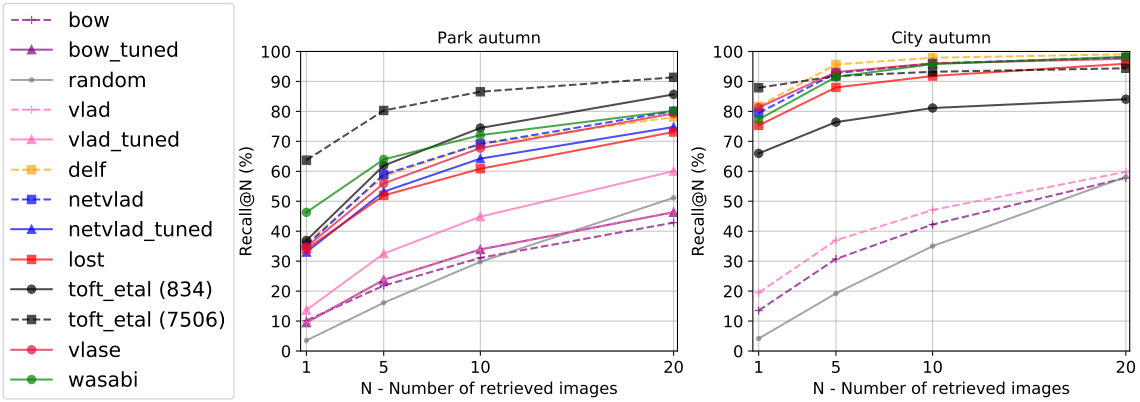


Figure 3.8: CMU-Seasons during Autumn with various weathers. The reference traversal is sampled during the winter.

Various Weathers in Autumn. Fig. 3.8 previously compares the retrieval scores averaged over several autumn traversals. The main source of appearance variations in the images are caused by the light. Although the park performances in autumn are slightly lower than for the spring, this is enough to suggest that recognition is harder in autumn than in spring. One explanation for the slight drop is, once again, a sun glare present in one of the autumn traversals. This induces a score drop of 10% in average over all methods. For the other traversals, the light induces a relative change of 1% only. Surprisingly, one of the city traversals also exhibits a performance drop. A qualitative observation shows that the main reason is that it is sampled when the sun is lower than in other traversals, which induces a global change in the scene colors. This comforts the previous assumption that illumination variations may have a higher impact than the ones induced by seasonal changes. The right plot (Fig. 3.8) suggests that current methods all have the same robustness to such variations.

Correlation between Season and Weather variations. As stated previously, light variations are a challenge in addition to the seasonal ones. Figure 3.9 shows the scene

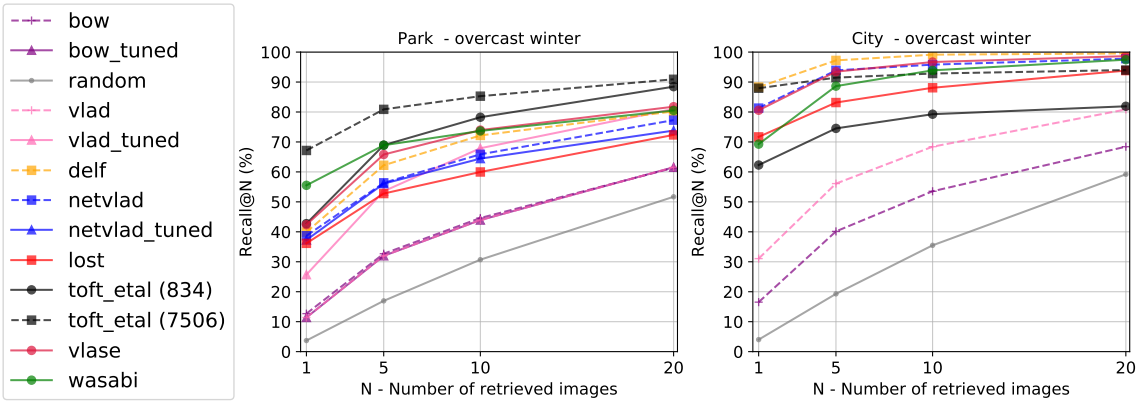


Figure 3.9: CMU-Seasons in Winter. The reference traversal is recorded during a sunny winter day whereas the queries are sampled during a day with overcast weather.

recognition performance over the same season as the reference traversal, winter, but with different illuminations. The reference traversal is sampled during a sunny day whereas the query traversal is overcast. Surprisingly, the performances on the park are equivalent to the autumn retrieval scores. One could have expected that the higher similarity between winter traversals would boost the results. This reinforces the assumption that illumination variations may be as challenging as the seasonal ones.

3.3.3 Robustness to semantic variations

	City						Park							
Slices	6		7		8		22		23		24		25	
Conditions	c0	c1	c0	c1	c0	c1	c0	c1	c0	c1	c0	c1	c0	c1
Dense trees								x	x	x		x		
Sparse trees				x			x				x			
Grass				x			x				x			
Road	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Sidewalk	x	x	x	x	x	x		x					x	x
Buildings	x	x	x	x	x	x							x	x

Table 3.7: Semantic elements in each slice. Note that the park slice 25 holds urban elements, camera 1 from slice 7 captures scenes with natural elements, and slice 23 display less features than other slices with dense trees occupying roughly 80% of the images.

Rationale. Another way to split the CMU dataset is to divide it into the contiguous slices based on the car's position, as in [157]. This adds to the natural urban/park separation. Each slice holds images with specific semantic structures. For example, the

park slice 25 holds mostly bridge and building skyline elements that fall into the urban semantic category rather than the bucolic one. Evaluating scenes with respect to the slice content over all season-light conditions amounts to evaluating the scene recognition with respect to the semantics. Table 3.7 details the slices peculiarities. The main takeaways of this table are that although captured in the city, images sampled over slice 7 with camera 1 hold bucolic elements. Also, some park slices display less semantic content than others such as the right cameras of slice 22 and 24 as well as the whole slice 23. For these slices, the camera capture scenes with mostly dense trees occupying approximately 80% of the scene.

Results. Overall, WASABI exhibits a significant advantage over SoA on scenes with sparse bucolic elements. It is limited by the amount of geometric information available: when the images have dense foliage, the performance drops to the level of existing methods. On urban environments, it compares to NetVLAD and DELF although it is not specifically tailored for it. This shows that although hand-designed, the WASABI visual features can generalize.

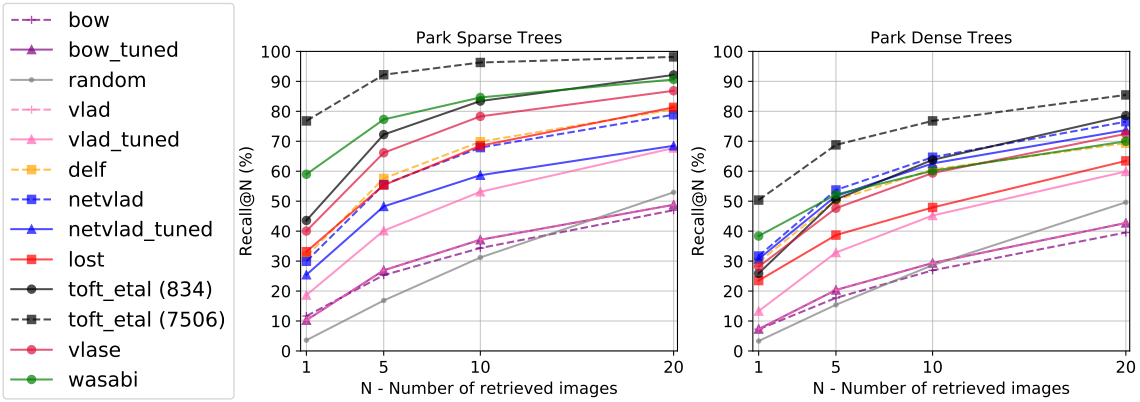


Figure 3.10: Ext-CMU-Seasons. Retrieval results on scenes with sparse vegetation *v.s* scenes with dense vegetation. All methods exhibit a strong sensitivity to dense vegetation: the main challenges are the few semantic edges used by semantic approaches, and the repetitive patterns used by pixel-intensity-based approaches.

Bucolic scenes with sparse foliage. Figure 3.10 shows the results for park slices with sparse trees along the road. WASABI exhibits a significant advantage over the others, especially when the error tolerance is small (*Recall*@N1).

Bucolic Scenes with dense Foliage. When the slices hold mostly dense trees along the road, all performances drop (Figure 3.10). The images not only have few features but also few semantic edges on which WASABI relies. This limit motivates the exploitation of multiple scales of edge information and not only the coarse one. Note that finetuning NetVLAD exceptionally proves to be relevant for slice23. This suggests that retrieval could be learned in challenging bucolic environments. However, the average performance on the remaining slices suggests that a simple finetuning may not be enough.

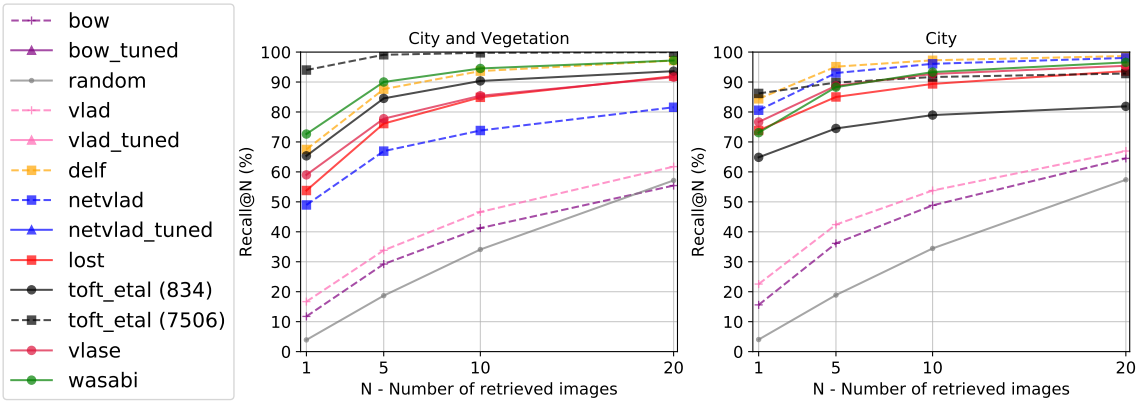


Figure 3.11: Ext-CMU-Seasons. Retrieval results on urban scenes with vegetation elements *v.s.* urban scenes with only city structures.

Urban Scenes. WASABI compares with SoA NetVLAD scores on urban scenes even though it is not specifically tailored for such environments (Figure 3.11). It is interesting to note that when there are vegetation distractors along with the urban structures (left plot), WASABI slightly outperforms SoA urban approaches. These scenes mostly hold grass and trees along a parking lot instead of buildings only. This observation supports the bias that existing methods have toward urban environments.

3.3.4 Global evaluation on Symphony.

Rationale. The Symphony dataset captures scenes with the same semantic content all along the shore with season and light conditions randomly sampled among 141 traversals collected every two weeks for three years. The wide range of season and light condition it holds make it suitable for a global evaluation of scene recognition over several conditions.

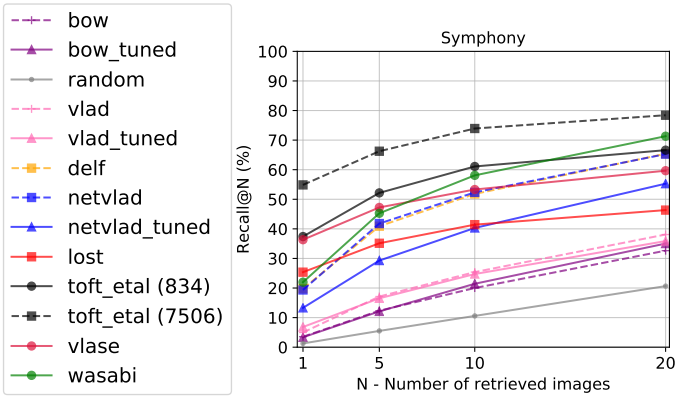


Figure 3.12: Symphony global performance measured through *Recall*@N. WASABI only compares to the SoA tailored for urban environments but falls behind VLASE and [182]. One explanation is that the segmentation is too noisy on the Symphony dataset. This noise is propagated in the image description.

Results. WASABI presents a slight advantage over NetVLAD and DELF although one could have expected higher performance based on the previous solid results on CMU-Seasons. One explanation is the segmentation noise induced by the image noise in one hand (*e.g.*, sun glare) and the lack of domain adaptation on the other hand (Fig. 3.6). As there is no ground-truth segmentation for the Symphony dataset, finetuning the segmentation is currently not possible. However, the satisfying results on CMU-Seasons motivate future work to improve the Symphony segmentation as well as the robustness of the descriptor to failures of the segmentation stage.

3.4 Summary

This chapter introduced a novel image global descriptor from the fusion of hand-design and machine learning to make it robust to long-term variations in visual appearance. Provided a semantic segmentation CNN is available, the wavelet description of semantic edges is aggregated into a descriptor. When tested on multi-season bucolic datasets, it achieves or surpasses the SoA. It even compares to them in urban settings for which the SoA is specifically tailored for. This highlights that WASABI can generalize and does not need manual tuning. This suggests that fusing hand-design and machine learning can be a solution to the generalization challenge of learning methods.

One limit of the current method is the use of only coarse hand-picked useful locations. So WASABI can not handle scenes with dense features or low-scale information. This motivates the next chapter to define an approach that also includes useful locations

at a pixel-wise scale.

Chapter 4

Local Feature for Long-Term Visual Scene Recognition

The previous chapter introduced a novel global descriptor from the fusion of semantics and geometry. It proves robust to long-term variations in visual appearance but only leverages coarse-scale information. This is especially limiting for scenes with small and dense elements, or when the semantic edges are noisy. This motivates this chapter to select image features at finer scales and aggregate them in a VLAD fashion (Fig. 4.1). Keypoints are acceleration maxima of the semantic edges. They are described with the Shape Context Descriptor [22], which represents the local distribution of the edge points around that location. These local features are then fused into a global image descriptor with a modified VLAD aggregation, where local features are assigned to visual words with the same semantic label. Experiments show that it boosts the retrieval performances on the challenging Symphony dataset. Another improvement over WASABI is that the image description is now more compact. It is even four times smaller than NetVLAD and DELF before their dimensionality reduction step. Also, the image distance is a simple Euclidean distance between their descriptors.

As in BOW and VLAD, the visual word index is generated by clustering local features computed on a set of training images. The centroids, that is the clusters' centers, are the visual words. A slight modification from the standard BOW/VLAD clustering is introduced to better leverage the semantic information: the local features are first grouped by class before being clustered. This way, a visual word has a semantic class.

The VLAD aggregation is also modified to integrate semantic information. The local features extracted from an image are associated with the nearest visual word with the same semantic class. The residues of these associations are accumulated for each word and concatenated to produce the global image descriptor. It has the shape $d * \mathrm{N}_d$ with d the local descriptor dimension, and N_d the number of visual words.

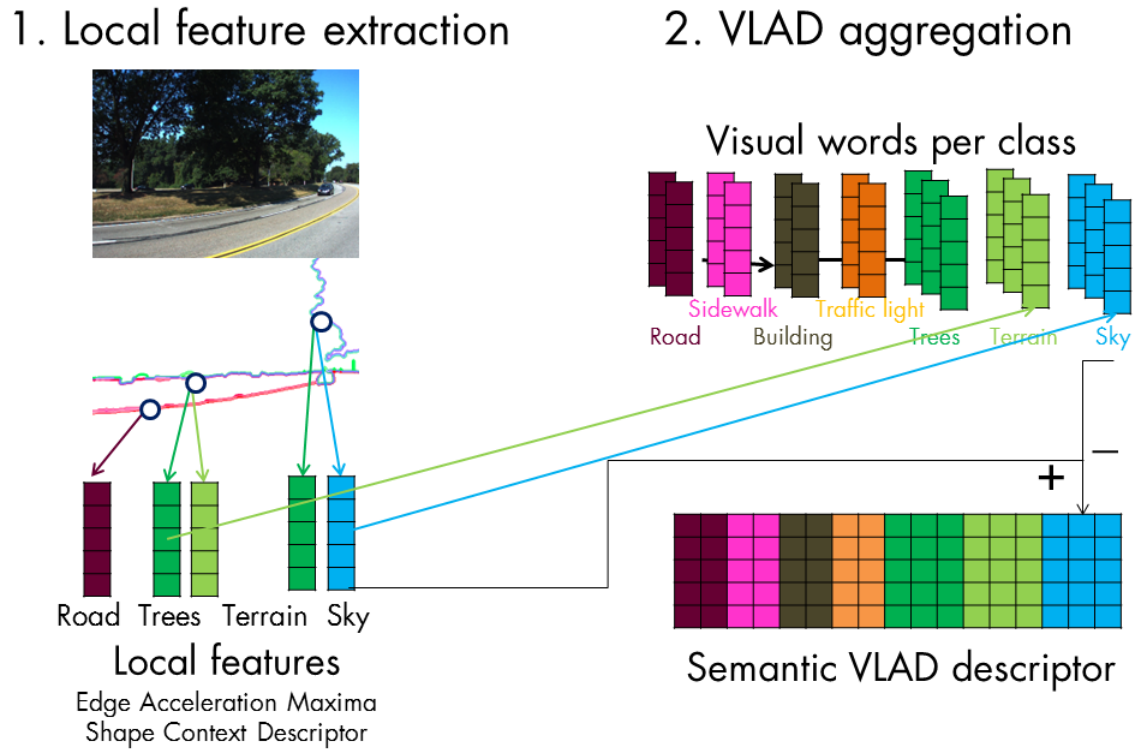


Figure 4.1: Illustration of the global description. Local features are located along semantic edges where the edge acceleration is maximum. They are described with the local edge variation derived with the Shape Context Descriptor [22]. Local features are separated according to their semantic labels. A visual codebook is computed by clustering these local features on a training dataset. The clusters are derived within groups of local features with the same labels. A query image is described by aggregating its local features in a semantic VLAD fashion where each feature is associated with the visual word with the same class.

The rest of this chapter is organized as follows. Sec. 4.1 reviews the limits of WASABI introduced in the previous chapter. Sec. 4.2 then details the novel multi-scale approach derived in this chapter. Experiments in Sec. 4.3 show that this novel approach compares to SoA performances on bucolic and urban scenes, and even doubles it over the challenging Symphony dataset.

4.1 Limits of Coarse Features for Scene Recognition

This section reviews the limits of the WASABi descriptor, introduced in the previous chapter. WASABI selects semantic edges as relevant locations and describes them with

their wavelet transform [41] over a fixed-sized subsampling of the edge. When used for scene recognition, these features are aggregated by simply concatenating their descriptors. The resulting vector provides a global description of the image. To compute the distance between two images, edges from one image are associated with the nearest edge with the same semantic label in the other image. The distance between the two edges is the Euclidean distance between descriptors, and the image distance is the sum of the distances between associated edges.

WASABI exhibits the following limits:

1. *It relies only on a coarse edge description.* It ignores the edge's local variations that can be used to further characterize the edge.

2. *The WASABI image representation is not scalable* with the number of edges. An image is represented by the collection of descriptors for each semantic edges. So the size of this global representation increases with the number of semantic edge.

3. *The image distance complexity increases quadratically* with the number of edges. The distance between the two images is the sum of the distance between their matching edge descriptors. Although successful, a better distance computation would avoid the loop over all edges. Despite these restrictions, WASABI demonstrates that fusing hand-designed geometric information with learned semantic is a relevant approach to define visual features robust to long-term variations.

4.2 Scene Recognition from Local Hand-Crafted Features

This section derives a new image descriptor dubbed SG-VLAD [1] [2], for Semantic and Geometric VLAD, to address WASABI's limits.

4.2.1 Local Keypoints: Semantic Edge Acceleration Maxima

A keypoint is a point on a semantic edge that is also an edge acceleration maximum. With the edge framed as a parametric curve $(x(t), y(t))$, the acceleration is given by $(\frac{d^2x}{dt^2}, \frac{d^2y}{dt^2})$. For the two 1D signals $x(t)$ and $y(t)$, the second-order derivative is the Laplacian. To avoid numerical edge-cases and accelerate the computation, the Laplacians are approximated with a Difference of Gaussians (DoG), as for SIFT. The multiplication factor between the standard deviation of the two Gaussians is 1.6 to best approximate the

[1] https://youtu.be/JeYpcRPqDUM
[2] https://github.com/W2desc/wasabi2.git

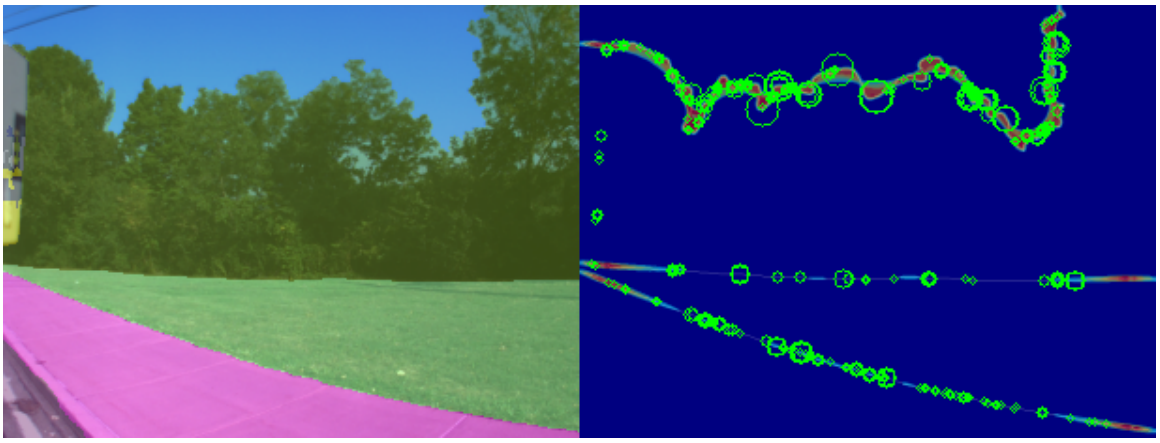


Figure 4.2: Edge acceleration heatmap with detected keypoints drawn in black circles. The size of the circle represents the scale of the keypoint.

Laplacian. Local acceleration maxima are computed in both space and scale. The standard deviation of the Gaussian used to derive the scale-space representation of the edge is also 1.6. Edge maxima are located with sub-pixel accuracy by fitting a 2D quadratic function to the local acceleration and finding its root. It is the 2D Euclidean analog to the sub-pixel refinement used in SIFT [116]. A keypoint is defined by a triplet (x, y, s) where (x, y) are the point location and s the edge scale for which this point is an acceleration maximum. The point local orientation is not computed at this stage as this information is present in the descriptor described in the next paragraph.

The reader familiar with feature detectors based on curvature maxima may wonder over the advantages of this acceleration approach. Curvature-based methods are akin to corner detection on the edge. Given the geometric nature of the edge, this detector activates on much fewer locations. This weakens the final image description that relies only on sparse local geometric information. Hence the motivation for the acceleration-based approach that provides more repeatable keypoints to exploit.

4.2.2 Local feature description

Keypoints are described with the Shape Context Descriptor (SCD) introduced by Belongie *et al.* [22]. It describes a point on an edge with the 2D histogram of directions between this point and its neighbors (Figure 4.3). This naturally captures the point's local orientation with respect to its neighbors. This is why it is not deemed necessary to explicitly compute the keypoint orientation in the detection step.

The SCD parameters are the histogram discretization. In all experiments, the orientation is discretized over 12 bins covering 360°. Given the keypoint scale s, the distance

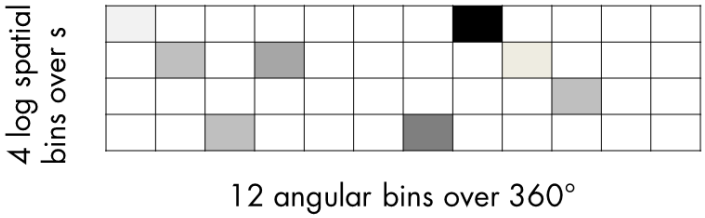


Figure 4.3: Shape Context Descriptor 2D histogram [22].

between the keypoint and its neighbors is discretized over 4 bins covering a distance up to $log(s)$. The log-sampling is a way to give more importance to nearer edge variations. Finally, the histogram is flattened to get a $12 \times 4 = 48$-dimension descriptor.

4.2.3 Semantic Codebook

Semantic visual words are derived by clustering local features sampled from training images. In the standard codebook derivation, the clusters are computed over all features. Instead, here, a set of clusters is computed over local features in each semantic class. The number of clusters is specific to each semantic class, which presents two advantages.

The first one is that it provides additional information on the features that serve to better discriminate them. This is especially relevant for the edge description as edges can be locally similar so semantics better separate them.

The second is that it allows modulating the importance of each class in the image description. For example, urban structures are much more important than vegetation for scene recognition in the city. So it is intuitive to allocate more visual words to index urban features than vegetation features. The reverse strategy is adopted when most of the scenes depict bucolic environments, as in the CMU park.

4.2.4 Semantic Aggregation

Local features are aggregated in a VLAD fashion in which semantic constraints are integrated.

The formal standard aggregation is recalled here. Let $d \in \mathbb{N}$ be the local feature's dimension, $(v_i)_{i \in [\![0, \mathrm{N}_d - 1]\!]}$ be N_d visual words of dimension d, and let $(f_j)_{j \in [\![0, \mathrm{N}-1]\!]}$ be N local features of dimension d sampled over the image. The VLAD descriptor is derived by accumulating the residuals between each local feature and its nearest visual word. More formally, it is a 2D matrix $\mathrm{V} \in \mathbb{R}^{\mathrm{N}_d \times d}$ where $\mathrm{V}(r, c) = \sum_{j=0}^{\mathrm{N}-1} \delta_{r,j}(v_r[c] - f_j[c])$ with $\delta_{r,j} = 1$ when the j^{th} local feature is assigned to the r^{th} visual word, and 0 else way.

Intuitively, the r^{th} line stores the cumulative residues between the r^{th} visual word and the local features that are assigned to it. The 2D matrix is flattened and the descriptor is normalized with Signed Square Rooting (SSR) following the guidelines in [79, 82], *i.e.* the descriptor V becomes $sign(\mathrm{V})\sqrt{|\mathrm{V}|}$.

SG-VLAD modifies the aggregation to assign the local feature f_j to the nearest visual word v_i with the same semantic label.

4.2.5 Addressing the Coarse Approach's Limits

1. *WASABI relies only on a coarse edge description.* An intuitive assumption is that the more informative the visual features are, the more useful they are for the end-goal task. By leveraging local information, SG-VLAD computes richer image representation. This is particularly relevant for images with few semantic elements: there are few edges to exploit whereas there can be many local edge variations specific to the image. For example, in bucolic scene visual recognition, the skyline between the sky and trees is usually the most informative part of the image. Although most skylines look globally similar, the local variations are highly discriminative.

2. *The image representation is not scalable with the number of edges.* SG-VLAD's image descriptor is a fixed-sized $48 * \mathrm{N}_d$ vector. It depends neither on the number of edges nor on the number of local features.

3. *The image distance is not scalable with the number of edges.* Now that an image is described with a simple 1D vector, computing the image distance is a simple $\mathscr{L}_2$ norm of the vectors' difference.

4.3 Experiments

The finer features of SG-VLAD reach similar or better performances than WASABI and SoA deep learning approaches. This shows that semantic and geometric fusion for image description can be a relevant alternative to end-to-end approaches when heavy training and large specific datasets are not feasible. The rest of this section details the experimental results.

Datasets and Metrics As for WASABI, the retrieval performance is computed on the CMU-Seasons and the Symphony datasets. The CMU-Seasons dataset evaluates the scene recognition with respect to semantics variations in one hand, and season-light conditions on the other. The Symphony dataset measures the global performance of

SG-VLAD over a wide range of season and illumination variations of scenes dominated by vegetation. Performances are measured with the *Recall@*N.

Setup All GPU-based methods are run on a NVIDIA 1080Ti with Tensorflow 1.12, Cuda9, Cudnn7 and OpenCV 3.4. The CPU-based methods are run with Python3 or C++. More details are available in the released code. The baselines setup is the same as in the previous chapter. And the segmentation is computed with the same model provided by [103].

SG-VLAD's codebooks are generated by clustering local features on all images from the CMU park training slices {18-21} and 1230 images from the west side of the Symphony lake. These are the same image numbers used to generate the codebooks for VLASE. This way, it is possible to compare the data needs of each of these methods. Note however that, due to a code typo, the codebooks for `bow_tuned` and `vlad_tuned` were trained only on 1691 CMU park images.

Class	CMU-Park	CMU-City	Symphony	Class	CMU-Park	CMU-City	Symphony
Road	2	4	2	Traffic Light	2	4	0
Sidewalk	2	4	1	Traffic Sign	2	4	0
Building	2	4	1	Vegetation	4	2	4
Wall	2	4	0	Terrain Sky	4	2	4
Fence	2	4	0	Sky	4	2	4
Pole	2	4	1	-	-	-	-

Table 4.1: Distribution of the number of visual words per semantic class on CMU-Seasons (CMU) and Symphony (SYM).

Only semantic edges from non-moving classes are kept. The number of clusters per class is summarised in Table 4.1: the general approach is to assign more words to classes that are deemed more important or more represented. Some urban elements in the CMU park are still present so they are allocated two visual words. Four visual words are assigned to nature-related classes such as vegetation or terrain. Four words are also used for the skyline as it usually adjacent to the vegetation. The inverse distribution is adopted for the CMU city. For the Symphony dataset, some urban elements are absent or too infrequent so they are not represented in the codebook. Note that the road label with class id 0 is replaced with a 'water' label. In total, we use $N_d = 39$ visual words on the park slices and $N_d = 17$ visual words on the lake. Further experiments on the semantic distribution of the visual words are ongoing.

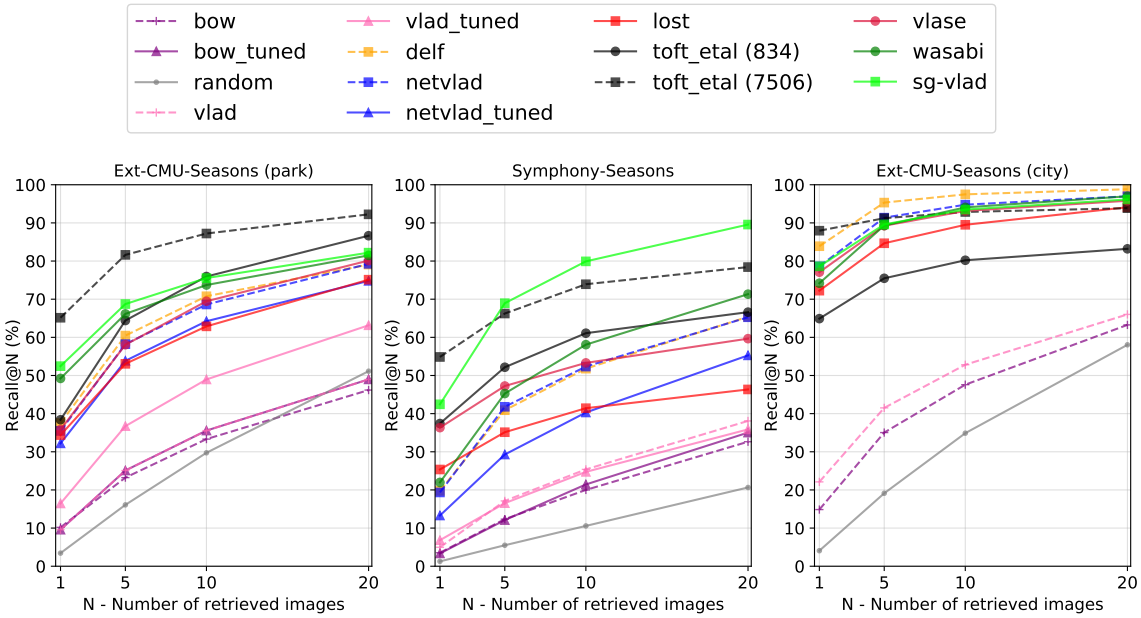


Figure 4.4: SG-VLAD improves over WASABI on the Symphony dataset and achieves similar results to VLASE and the low-resolution of Toft's descriptor [182]. It preserves the performance on the CMU scenes while addressing the scalability limits of WASABI.

4.3.1 Global evaluation on Extended-CMU-Seasons

Fig. 4.4 compares the new SG-VLAD descriptors to the other methods described in the previous chapter. While it achieves similar performance as WASABI on the CMU dataset, it significantly improves on the Symphony retrieval and achieves scores comparable to VLASE and the low resolution version of [182] (toft-834). This agrees with the previous observation that methods that leverages local semantic information seem more robust to noisy segmentation, such as the Symphony one. One possible explanation is that histograms of either pixel labels or local semantics seem to be invariant to such amount of noise. Another motivation to favor SG-VLAD over WASABI is that the image descriptor has now a fixed size and the image similarity is assessed with the descriptor Euclidean distance. The rest of this section provides further analysis of SG-VLAD robustness, as was done for WASABI.

4.3.2 Robustness to Illumination Variations

Rationale. We start with a reminder of how the illumination robustness is evaluated. Each CMU traversal captures scenes from urban and bucolic environments with a specific season and light condition, for example, overcast-winter, sunny-autumn. So, the scene recognition performance over one complete traversal amounts to evaluate the

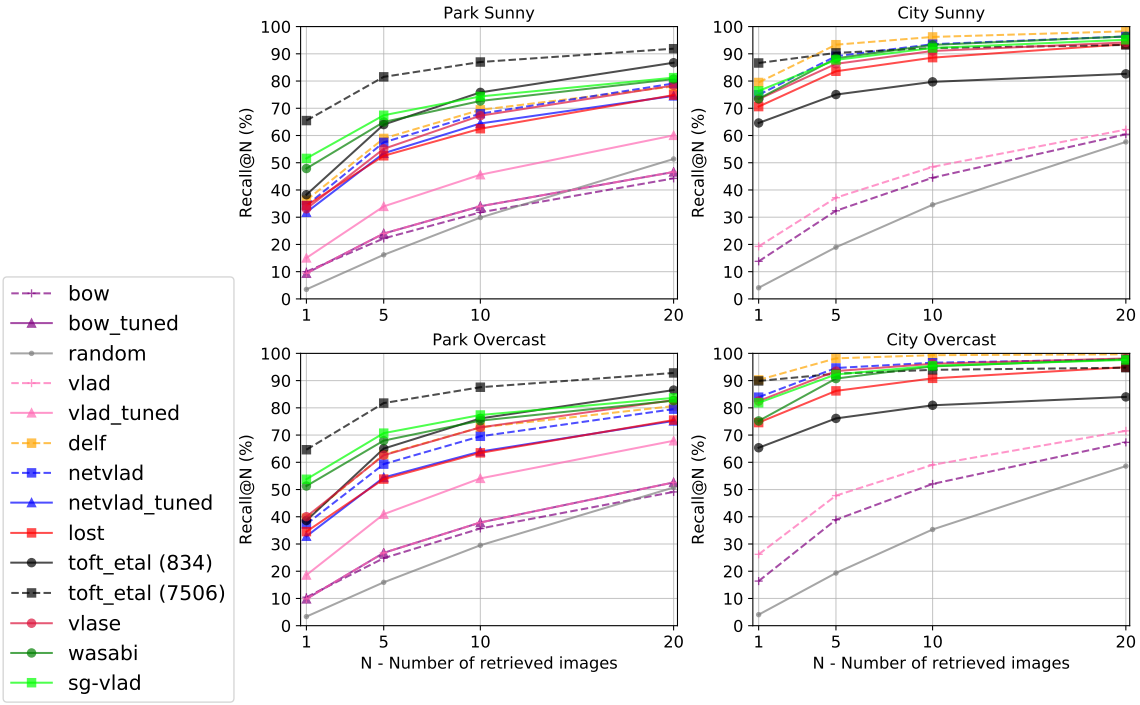


Figure 4.5: Retrieval scores grouped by light conditions: sunny (top), overcast (bottom). Overall, the *Recall*@N for overcast scenes is 10% higher on average than for the sunny scenes. This is expected given that the reference traversal was sampled during an overcast winter. When the light varies, SG-VLAD's performance evolves the same way as existing approaches, which suggests that it is as robust to light variations.

performance with respect to one-season light condition. Here, traversals with the same illumination are evaluated together. This provides an observation of the light's influence on the retrieval.

Results. Globally, SG-VLAD evolves the same way as existing methods when the illumination changes (Figure 4.5). This suggests that it is robust to light variations as the SoA.

The performance variations from one illumination to another change differently for each method, which prevents drawing a trend on the light's influence on the retrieval. Overall, the numerical gaps are lower than 10%, with some descriptors exhibiting stronger shifts than others. For example, the WASABI and the Toft *et al.* [182] descriptors vary the least, closely followed by SG-VLAD and NetVLAD. Surprisingly, the DELF score on the overcast city retrieval is 10% higher than in the sunny city. This is unexpected as deep features usually describe the image with pixel statistics high enough to ignore such low-level intensity variations.

Note that the evaluation ignores the seasonal variations in appearance for the park retrieval. Indeed, it evaluates the *Recall*@N with respect to the light only. The same is done for the city scenes where it is reasonable to assume that the image content did not significantly change. Still, the performance shifts respective to each method are analog across both environments. This suggests that the correlation between the retrieval performance and the illumination variations may be tackled independently of the seasonal variations.

4.3.3 Robustness to Semantic Variations

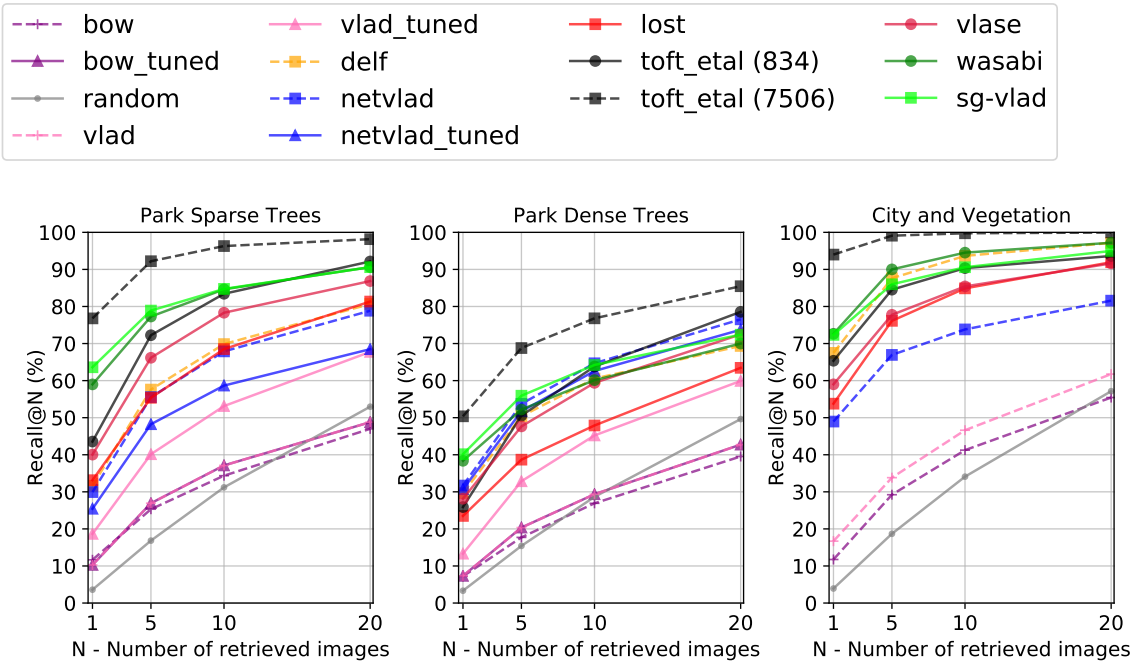


Figure 4.6: Ext-CMU-Seasons: Retrieval performance with respect to the semantic content of the images. SG-VLAD presents a slight advantage over most methods when the scenes hold vegetation elements.

Rationale. The reader is reminded of the evaluation's motivation detailed in the previous chapter for WASABI. Each CMU traversal is split into continuous slices based on the car's position. So each slice covers scenes with the same semantic structures over long periods. Evaluation on one slice amounts to evaluating this task with respect the slice's semantics. Slices with similar semantic content are evaluated together.

Results SG-VLAD presents the same advantages as WASABI for retrieval on scenes with bucolic content while offering better scalability properties (Fig. 4.6). However, contrary to what was expected, leveraging local edge information instead of the coarse one does improve the performance when the vegetation is dense. The rest of this paragraph further details the results.

Bucolic scenes with sparse foliage. When the scenes hold sparse vegetation elements (Figure 4.6 - left), leveraging finer features, as done in SG-VLAD, does not boost the results over WASABI. One reason is that when natural elements are sparsely distributed in the image, coarse edge information is already informative enough to describe the image. SG-VLAD still has the advantage of a shorter and fixed-size image descriptor as opposed to the non-scalable WASABI one.

Bucolic scenes with dense foliage One of the motivations for SG-VLAD was to tackle the current recognition limits when the scene holds dense vegetation. The main challenges come from the repetitive pixel intensity patterns, and the few edges and the little semantic information. This explains why the *Recall*@N drops for most methods. WASABI and SG-VLAD were designed to leverage the little information that discriminates between such images, mostly the geometric information of semantic edges. WASABI slightly improved over existing approaches by exploiting the global edge geometry, but it could be improved. So this motivated the design of SG-VLAD to leverage additional information in the local edge variations. However, experiments show that it does not induce the expected boost. Investigating further improvements, including the exploitation of finer edges than the semantic ones, is the object of future work.

Note that contrary to images with sparse vegetation, VLASE and Toft *et al.* [182] descriptors reach the same performance as non-semantic methods. Although the gap is of only 10%, it shows the benefit of leveraging both the geometric and the semantic information in SG-VLAD. VLASE is made of the fusion of semantic probabilities over the edge, and while it offers robustness to illumination and seasonal variations, it underperforms on the dense vegetation scenes. In addition to the few edges to leverage, another reason may be that the highly repetitive patterns in these scenes lead to similar semantic edge probabilities. So their aggregation into an image descriptor is not discriminative enough to differentiate such scenes. A similar explanation holds for the [182] descriptor of which both the semantic histogram and SIFT-like descriptors are similar across images. Once again, the main cause is the repetitiveness of the image information.

Note that this is also an open problem for urban environments. One of the few works that tackle this specific problem is [188]. Torii *et al.* propose to weight the aggregation of local features so that repetitive ones do not dominate the sparser one. However, this processing can not be integrated as is since dense bucolic scenes are entirely dominated by repetitive patterns. So there is no other discriminative information to balance them with. The integration of such balancing is the object of future work to tackle the challenge of repetitive patterns in natural scenes.

Urban Scenes. (Figure 4.6) shows that SG-VLAD does generalize to other scenes than the bucolic ones. This is a significant advantage over SoA approaches. Note that the descriptor dimension here is significantly smaller than SoA based approaches before their dimensionality reduction step: 48 for WASABI *v.s.* 256 for NetVLAD and 1024 for DELF.

4.3.4 Global evaluation on Symphony.

Rationale. As stated in the previous chapter, the Symphony dataset captures scenes with the same semantic content with various seasons and light conditions. The wide range of season and light condition it holds make it suitable for a global evaluation of scene recognition over several conditions.

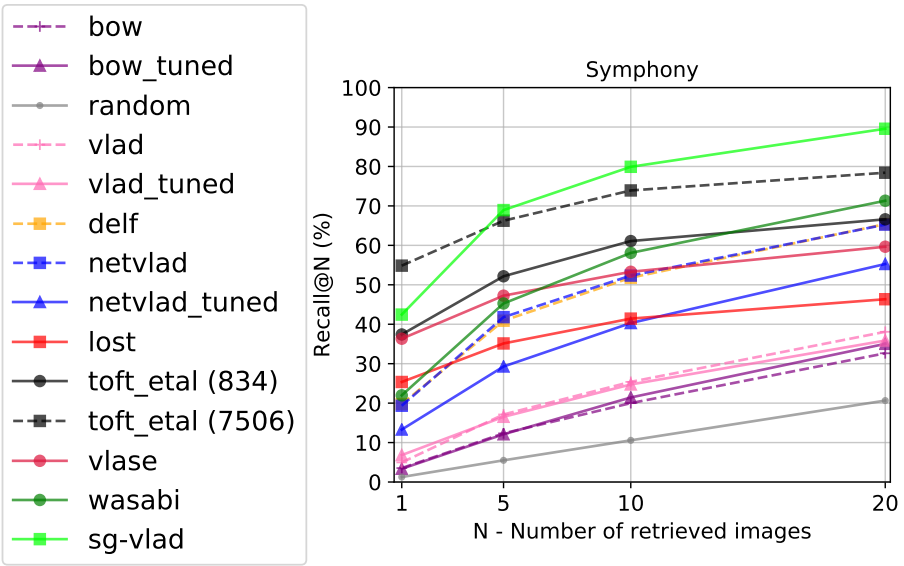


Figure 4.7: Symphony global performance. SG-VLAD increases by 100% over the previous SoA.

Results. Figure 4.7 shows that SG-VLAD doubles the results previously reached by WASABI. This result is even more outstanding given that this environment presents much harder challenges than the CMU park. A possible explanation for this significant improvement is that Symphony often exhibits few semantic edges per image. If an image holds only 2 edges, WASABI represents it with only 2 coarse descriptors. But SG-VLAD uses local information that is more present so the descriptor is richer and easier to discriminate. Future work will build upon SG-VLAD to exploiting edges other than the semantic ones.

4.4 Summary

This chapter presented SG-VLAD, a novel image descriptor based on semantics and geometry to overcome the limits of its predecessor, WASABI, that only leveraged coarse-scale information. Provided with the same semantic segmentation, it adopts an approach analog to VLAD and aggregates local geometric keypoints. A keypoint is an acceleration maximum of one semantic edge. It is described with the local distribution of the edge at that point. These local features are aggregated in a VLAD fashion while integrating semantic information.

Experiments show that it reaches SoA on bucolic environments and even induce a 100% performance improvement on the extremely challenging Symphony dataset. It compares to deep-learning SoA on urban environments even though it required no deep training and exploits much more compact representations. These results reiterate the relevance of hand-design and machine learning for useful visual features.

One limit that it shares with WASABI is the sensitivity to segmentation noise. Another limit is that it does not address the open problem of describing repetitive patterns. This is a problem also common to urban settings for which a reliable solution is yet to come. Finally, while semantic edges usually carry enough information to discriminate between images, leveraging other types of edges could prove useful. These three lines of study are the object of future work.

Part II

Unsupervised Local Features from Trained CNNs

The previous parts addressed the problem of robustness in scene recognition by manually fusing semantics and geometry to design novel image global descriptors. Experiments showed that this approach is as performant as a heavily trained deep learning approach on bucolic scenes and reaches reasonable results in urban environments.

This part now turns on the problem of reducing the learning overhead when defining data-specific *local* features. The proposed solution draws motivation from the edifying observations in [57, 113]. Long *et al.* [113] discusses how the representation space generated by a CNN is correlated to the image space with a 'high' resolution. Fischer *et al.* [57] show that this same space is discriminative enough to provide local descriptions for image regions. The next chapter continues efforts towards this research direction: it defines a local detector and a descriptor from a trained CNN that is as performant on image matching as when the CNN is specifically trained for this task.

Chapter 5

Local Features from pre-trained CNN [1]

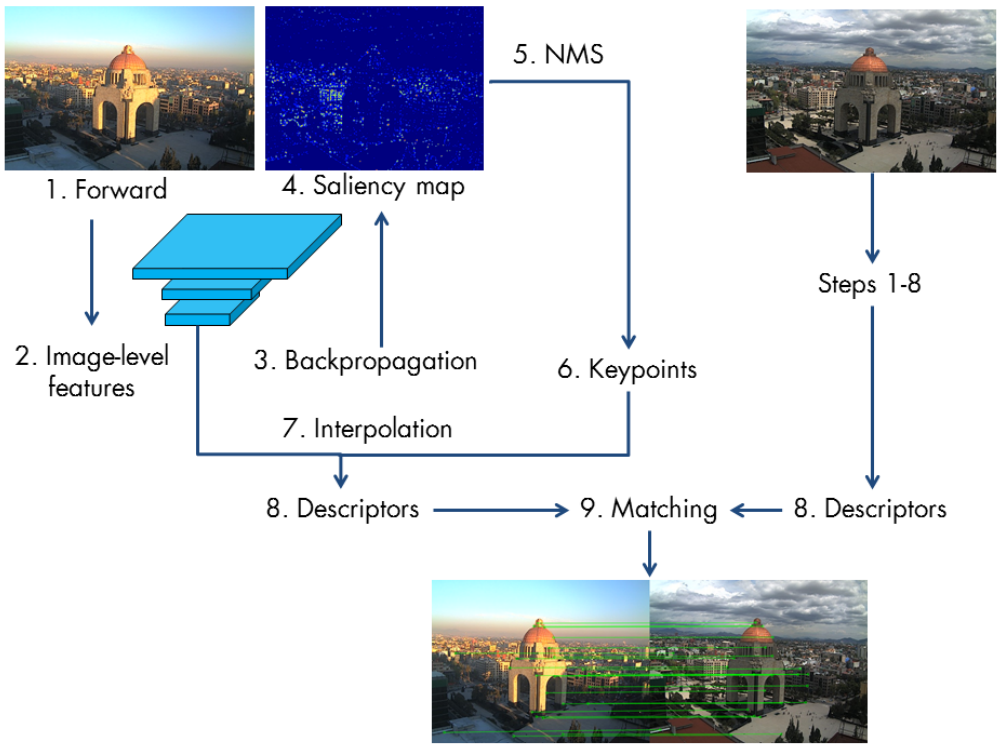


Figure 5.1: (1-6) Embedded Detector: Given a CNN trained on a standard vision task (classification), we backpropagate the feature map norm back to the image space to compute a saliency map. It is thresholded to keep only the most informative signal and keypoints are the local maxima. (7-8): Local descriptors are computed from the feature map interpolation on the detected keypoints.

This chapter introduces a novel feature detector from the information embedded inside a CNN already trained on standard learning tasks, such as classification, with no fur-

[1] This chapter describes contributions published in ICCV 2019 [24].

ther training. Fig. 5.1 illustrates the method dubbed ELF (Embedded Localization information from CNN Features). A score map is computed from the gradient of the feature map norm with respect to the input image. The local maxima of this map are relevant keypoints. Note that contrary to recent deep learning methods, this does not require to train the CNN. These keypoints are as repeatable as the ones from hand-crafted and learned detectors. The same CNN is used to get local descriptors by interpolating one of its feature maps on the detected keypoints. The resulting local features achieve similar matching performance and robustness on standard evaluation datasets (HPatches [19], Webcam [191]). This shows that the representation space and the feature localization learned by a CNN to complete a visual task is relevant to define local features. And these features are as relevant as when the CNN is specifically trained to produce them.

The rest of this chapter is organized as follows. Sec. 5.1 recalls the related work relevant to this chapter. Sec. 5.2 and 5.3 detail the detector and descriptor derivations. Experiments in 5.4 show that it compares to the SoA matching performances.

5.1 Review of Local Features

This section summarizes the local features related to ELF. See Sec. 2.2 for more details.

Early methods rely on hand-crafted detection and description: SIFT [116] detects 3D spatial-scale keypoints on differences of Gaussians and describes them with a 3D Histogram Of Gradients (HOG). SURF [21] uses image integrals to speed up the previous detection and uses a sum of Haar wavelet responses for description. KAZE [8] extends the previous multi-scale approach by detecting features in non-linear scale spaces instead of the classic Gaussian ones. ORB [153] combines the FAST [151] detection, the BRIEF [32] description and improves them to make the pipeline scale and rotation invariant. MSER-based detector hand-crafts desired invariance properties for keypoints, and designs a fast algorithm to detect them [121]. Even though these hand-crafted methods have proven to be successful, they are now outperformed by learning-based methods.

One of the first learned detectors is TILDE [191], trained under drastic changes of light and weather on the Webcam dataset. It uses supervision to learn saliency maps of which maxima are keypoint locations. Ground-truth saliency maps are generated with 'good keypoints': it uses SIFT and filters out keypoints that are not repeated in more than 100 images. One drawback of this method is the need for supervision that relies on another detector. However, there is no universal explicit definition of what a good keypoint

is. This lack of specification inspires Quad-Networks [158] to adopt an unsupervised approach: they train a neural network to rank keypoints according to their robustness to random hand-crafted transformations. They keep the top/bottom quantile of the ranking as keypoints. ELF is similar in that it does not requires supervision but differs in that it does need to further train the CNN.

Recent learned detectors are trained within full feature extraction pipelines such as LIFT [199], SuperPoint [46] and LF-Net [135]. LIFT contribution lies in its original training method of three CNNs. The detector CNN learns a saliency map where the most salient points are keypoints. It then crops patches around these keypoints, computes their orientations and descriptors with two other CNNs. First, the descriptor is trained with patches around matching points with contrastive loss, then the orientation CNN together with the descriptor and finally with the detector. One drawback of this method is the need for ground-truth matching keypoints to initiate the training. In [46], the problem is avoided by pre-training the detector on a synthetic geometric dataset made of polygons on which they detect mostly corners. The detector is then finetuned during the descriptor training on image pairs from COCO [110] with synthetic homographies and the correspondence contrastive loss introduced in [40]. LF-Net relies on another type of supervision: it uses ground-truth camera poses and image depth maps that are easier to compute with laser or standard SfM than ground-truth matching keypoints. Its training pipeline is similar to LIFT but employs the projective camera model to project detected keypoints from one image to the other. These keypoint pairs form the ground-truth matching points to train the network. ELF differs in that the CNN model is already trained on a standard task. It then extracts the relevant information already embedded inside the network for local feature detection, which requires no training nor supervision.

A large literature studies how to train CNN to generate local descriptors: TFeat [20], MatchNet [69], DeepDesc [168]. These methods train the network to produce descriptor vectors with minimal/maximal similarity on matching/non-matching patches with either contrastive loss [68], triplet loss [20, 91] or correspondence-contrastive-loss [40]. Another approach learns task-specific descriptors for image retrieval for landmark recognition [133] or robot localization [156]. ELF breaks with this trend and builds on the experimental results of [57] to generate local descriptors from a trained CNN. The descriptor derivation used in ELF is not novel: the interpolation of a feature map on the keypoint location is a common practice, *e.g.*, UCN [40], SuperPoint [46], D2-net [51]. Contrary to these examples, ELF never finetunes the CNN previously trained on some visual task.

The ELF detection is inspired by the initial observation in [169]: given a CNN trained for classification, the gradient of a class score with respect to the image is the saliency map of the class object in the input image. A line of works aims at visualizing the CNN representation by inverting it into the image space through optimization [60, 117]. ELF differs in that it backpropagates the feature map itself and not a feature loss. The following works use these saliency maps to better understand the CNN training process and justify the CNN outputs. Efforts focus on the gradient definition [173, 174, 178, 206]. They differ in the way they handle the backpropagation of the non-linear units such as Relu. Grad-CAM [163] introduces a variant where they fuse several gradients of the classification score with respect to the feature maps and not the image space. Instead, ELF computes the gradient of the feature map and not a classification score with respect to the image. Also, it runs a simple backpropagation that differs from the guided backpropagation in the non-linearity handling. Finally, to the extent of my knowledge, this is the first work to exploit the localization information present in these gradients for local feature detection. See Sec. 2.4 for more details and feature gradients.

5.2 Low-level Feature Detection from CNN Saliency

This section describes ELF, a detection method valid for any trained CNN. Keypoints are local maxima of a saliency map generated from the gradient of one of the CNN feature map's norm with respect to the image. This map is automatically thresholded to keep only relevant locations using the data-adaptive Kapur threshold [90]. The remaining local maxima are the keypoints and their coordinates are computed with a simple NMS to get the maxima coordinates.

5.2.1 Saliency Score from CNN Feature Map

The saliency map S^l is a map that activates on the image regions that are the most informative for the l-th CNN feature map (Figure 5.2). It is computed as the gradient of the feature $\mathrm{F}^l(\mathrm{I})$ map with respect to the image I and evaluated on that same feature map. Another way to derive it is to backpropagate the feature map back into image space. From a geometrical point of view, this operation can be seen as projecting the gradient $\nabla_{\mathrm{I}}\mathrm{F}^l$ of the feature signal $\mathrm{F}^l(\mathbf{I})$ into the image space. From a signal processing approach, it amounts to filtering $\mathrm{F}^l(\mathbf{I})$ through $\nabla_{\mathrm{I}}\mathrm{F}^l$ into the image space.

More formally, let $\mathbf{I}$ be a vector image of dimension $\mathrm{D_I} = \mathrm{H_I} \cdot \mathrm{W_I} \cdot \mathrm{C_I}$, and F^l be a vectorized feature map of dimension $\mathrm{D_F} = \mathrm{H}_l \cdot \mathrm{W}_l \cdot \mathrm{C}_l$. The saliency map S^l, of dimension

D_I, is $S^l(\mathbf{I}) = \left|\nabla_I \left|F^l(I)\right|^2\right| = \left|{}^T F^l(\mathbf{I}) \cdot \nabla_I F^l\right|$, where $\nabla_I F^l$ is a $D_F \times D_I$ matrix and ${}^T F^l(\mathbf{I})$ is the transpose of the feature map vector.

Saliency activates on the image regions that contribute the most to the feature representation $F^l(\mathbf{I})$. The term $\nabla_I F^l$ explicits the correlation between the feature space of F^l and the image space in general. The multiplication by $F^l(\mathbf{I})$ applies the correlation to the features $F^l(\mathbf{I})$ specifically and generate a visualization in image space $S^l(\mathbf{I})$. If $C_I > 1$, S^l is converted into a grayscale image by averaging the saliency map across channels.

5.2.2 Feature Map Selection

The previous computation can provide saliency maps at different image scales depending on the feature map level l. High values of l correspond to deeper feature maps and generate coarse-scale utilities. The first levels generate maps where saliency focuses on low-level signals such as edges. Intermediate levels allow getting local features that can still capture high-level image representation. Currently, trial-and-error is the only approach to choose the relevant level l although saliency maps exhibit peculiar visual patterns that can reduce the search. This section details the experimental approach to choose the saliency level l.

The multi-scale property of such saliency comes from the CNN structure. CNN operations such as convolution and pooling increase the receptive field of feature maps while reducing their spatial dimensions. This means that F^l has less spatial resolution than F^{l-1} and the backpropagated signal S^l ends up more spread than S^{l-1}. This is similar to when an image is too enlarged as shown in Fig. 5.2. It shows the saliency computed from the $pool_2$ and $pool_3$ layers of the VGG [171] network. On the top row, $pool_2$'s gradient (Fig. 5.2-left) better captures the location details of the dome whereas $pool_3$'s gradient (Fig. 5.2-right) is more spread. Another consequence of this resolution loss is that small features are not embedded in F^l if l is too high. In that case, only coarse-level saliency can be recovered by the gradient computation. This would reduce the space of potential keypoint to only large-scale features which would hinder the image matching.

Remember that the SoA already provides ways to compute visual saliency at coarse levels. This chapter focuses on features at a higher resolution. This motivates the choice of the feature level l as the highest one that still provides pixel-wise feature localization. This is visually observable by a sparse high-intensity signal contrary to the blurry aspect of higher layers.

5.2.3 Automatic Data-Adaptive Thresholding

The map derived previously exhibits a peculiar distribution with sparse high saliency peaks. It is thresholded based on the saliency distribution to keep only the most useful locations. Figure 5.3 shows saliency maps before and after thresholding using Kapur's method [90], which is briefly recalled below. It chooses the threshold that maximizes the information between the image background and foreground *i.e.* the pixel distribution below and above the threshold. This method is especially relevant in this case as it aims at maintaining as much information on the distribution above the threshold as possible. This distribution describes the set of local saliency maxima.

More formally, for an image **I** of N pixels with n sorted gray levels, let $(n_i)_{i\in[\![0,n-1]\!]}$ be the corresponding histogram, *i.e.*, n_i is the number of pixels with gray level i. This defines an empirical pixel distribution where $p_i = \frac{n_i}{\mathrm{N}}$ is the probability of a pixel to hold the value n_i^{th} level. Let $s \in n$ be a threshold level and A, B the empirical background and foreground distributions: $\mathrm{A} = \left(\frac{p_i}{\sum_{i\leq s} pi}\right)_{i<s}$ and $\mathrm{B} = \left(\frac{p_i}{\sum_{i\geq s} pi}\right)_{i>s}$. The level s is chosen so that the two distributions A and B become independant. This is achieved by minimizing the mutual information I(A, B) between these distributions, which is equivalent to maximizing the sum of their entropy $\mathrm{H(A)+H(B)} = \mathrm{H(A,B)-I(A,B)}$. For better results, the saliency map is first denoised with a Gaussian of parameters $(\mu_{thr}, \sigma_{thr})$ before computing the threshold level.

Once the threshold is set, the saliency map is denoised a second time with a Gaussian of parameters $(\mu_{noise}, \sigma_{noise})$. Standard NMS (the same as for SuperPoint) iteratively selects decreasing global maxima while ensuring that their nearest neighbor distance is higher than the window $w_{\mathrm{NMS}} \in \mathbb{N}$. The keypoints within a distance of $b_{\mathrm{NMS}} \in \mathbb{N}$ pixels to the image border are ignored.

5.3 Local feature description from CNN

Inspired by SuperPoint's description, the keypoints are described by interpolating one the CNN feature map on the keypoints locations. The feature map level used for the description may be different from the one for detection. High-level feature maps have wider receptive fields so it is reasonable to assume that they embed more information than lower levels. This leads to more informative descriptors.

However, as the feature map level l increases, the feature map loses resolution. If the map level is too high, the interpolation of the descriptors generates vector too similar to each other. For example, the VGG $pool_4$ layer produces more discriminative descriptors than $pool_5$ even though $pool_5$ embeds information more relevant for classification.

Empirically, there exists a layer level l' above which the description performance stops increasing before decreasing. This is measured through the matching score metric introduced in [126]. The final choice of the feature map is done by testing a set of layers with increasing levels and select the lowest feature map before the descriptor performance stagnates.

5.4 Experiments

Baselines. This section compares the local features queried from the CNN against the SoA with available code: the fully hand-crafted SIFT [116], SURF [21], ORB [153], KAZE [8], the learning-based LIFT [199], SuperPoint [46], LF-Net [135], the individual detectors TILDE [191], MSER [121]. They are evaluated on how consistent the locations and their descriptions are across various light and viewpoint conditions.

OpenCV's code is used for SIFT, SURF, ORB, KAZE, MSER with the default parameters and the author's code is run for TILDE, LIFT, SuperPoint, LF-Net with the provided models and parameters.

Setup. The tests run on Nvidia QuadroM2200 and GeForce 1080Ti graphic cards, with Tensorflow 1.4, Cuda 8, Cudnn6 and Opencv3.4. The number of keypoints is limited to 500. All images are resized to the canonical size 480×640 px and the transformations are rectified accordingly.

The blurring parameters $(\mu_{thr}, \sigma_{thr})$, $(\mu_{noise}, \sigma_{noise})$ are set with a grid search in the range $[\![3,21]\!]^2$ and the NMS parameters $(w_{\text{NMS}}, b_{\text{NMS}})$ in $[\![4,13]\!]^2$.

Metrics. Standard validation guidelines [126] provide metrics to evaluate how consistent the useful locations and their descriptions are. The repeatability (*rep*) assesses how invariant the useful locations are. The matching score (*m.s.*) measures how discriminative the descriptors are. Given two images of the same scene taken under different light and viewpoint conditions, *rep* measures the percentage of locations common to both images. The *m.s.* is the percentage of corresponding locations for which descriptors are nearest neighbors.

As done in [46, 135], the overlap score used in [126] to compute correspondences is replaced with a 5-pixel distance threshold. Following [199], the *m.s.* is modified to include all descriptors in the greedy bipartite-graph matching. This differs from [126]

that includes only descriptors pairs of which inter-distance is below an arbitrary threshold. The latter's threshold may introduce bias as the descriptors have different dimensions. The rest of this paragraph details the metrics, that are also available in the released code [2].

Repeatability. Let $(\mathbf{I}^1, \mathbf{I}^2)$, be a pair of images and $\boldsymbol{KP}^i = (kp^i_j)_{j<\mathrm{N}_i}$ the set of N_i keypoints in image $\mathbf{I_i}$. Both metrics are in the range $[0,1]$ but are expressed as percentages for better expressibility.

Repeatability measures the percentage of keypoints common to both images. The locations in the first images are warped to the second one and the output coordinates are noted $\boldsymbol{KP}^{1,w}$. A naive definition of repeatability is to count the number of pairs $(kp^{1,w}, kp^2) \in \boldsymbol{KP}^{1,w} \times \boldsymbol{KP}^2$ such that $\|kp^{1,w} - kp^2\|_2 < \epsilon$, with ϵ a distance threshold. As pointed by [191], this definition overestimates the detection performance for two reasons: a keypoint close to several projections can be counted several times. Moreover, with a large enough number of keypoints, even simple random sampling can achieve high repeatability as the density of the keypoints becomes high.

The definition implemented in VLBench [105] solves this issue. It defines a weighted graph (V, E) where the edges are all the possible keypoint pairs between $\boldsymbol{KP}^{1,w}$ and $\boldsymbol{KP}^2$ and the weights are the Euclidean distance between keypoints.

$$\begin{aligned} \mathrm{V} &= (kp^{1,w} \in \boldsymbol{KP}^{1,w}) \cup (kp^2 \in \boldsymbol{KP}^2) \\ \mathrm{E} &= (kp^{1,w}, kp^2, \|kp^{1,w} - kp^2\|_2) \in \boldsymbol{KP}^{1,w} \times \boldsymbol{KP}^2 \times \mathbb{R} \end{aligned} \tag{5.1}$$

A greedy bipartite matching is run on the graph and matches are pairs with a distance less than ϵ_{kp}. With $\mathscr{M}$ be the resulting set of matches, *rep* is computed as:

$$repeatability = \frac{\#\mathscr{M}}{\min(\#\boldsymbol{KP}^1, \#\boldsymbol{KP}^2)} \tag{5.2}$$

Matching score. The matching score definition introduced in [126] captures the percentage of keypoint pairs that are nearest neighbors both in image space and in descriptor space, and for which these two distances are below their respective threshold ϵ_{kp} and ϵ_d. Let $\mathscr{M}$ be defined as set of keypoint matches based on their Euclidean distance, and $\mathscr{M}_d$ be the analog set of matches based of their descriptor distance instead. Keypoint pairs are deleted if their spatial (resp. descriptor) distance is above the thresholds ϵ (resp. ϵ_d). With $\#\mathscr{M} \cap \mathscr{M}_d$ the number of keypoints pairs which are both nearest neighbors in image space and descriptor space, *m.s.* is defined as:

[2] https://github.com/abenbihi/elf

$$matching\ score = \frac{\#\mathcal{M} \cap \mathcal{M}_d}{\min(\#\boldsymbol{KP}^1, \#\boldsymbol{KP}^2)} \tag{5.3}$$

Metrics parameters. The spatial distance threshold is set to $\epsilon = 5$, as is done in LIFT [199] and LF-Net [135]. Note that a way to reach perfect *rep* is to sample all the pixels or sample them with a frequency higher than the distance threshold ϵ_{kp} of the metric. One way to prevent this flaw is to limit the number of keypoints. Here the number of detected keypoints is limited to 500 for all methods.

As briefly described previously, one drawback of the *m.s.* definition is that there is no unique descriptor distance threshold ϵ_d valid for all methods. For example, the SIFT descriptor as computed by OpenCV is a $[0,255]^{128}$ vector for better computational precision, the SuperPoint descriptor is a $[0,1]^{256}$ vector and the ORB descriptor is a 32 bytes binary vector. Not only the vectors are not defined over the same normed space but their range varies significantly. To avoid introducing human bias by setting a descriptor distance threshold ϵ_d for each method, ϵ_d is set to ∞. This means that any descriptor match is valid as long as they match corresponding keypoints even when the descriptor distance is high.

Datasets. Various standard datasets allow for evaluation with respect to various conditions. Figure 5.4 shows examples from each set. The HPatches dataset gathers 116 image sequences with light and viewpoint variations. It is augmented with artificial scale and rotation transformations for further robustness analysis. The Webcam dataset displays static outdoor scenes with a wide range of natural light. Further details on each dataset are provided with the experimental results.

5.4.1 General performance

Dataset. The HPatches dataset [19] gathers a subset of standard evaluation images such as DTU and OxfordAffine [4, 125]: it provides a total of 696 images, 6 images for 116 scenes and the corresponding homographies between the images of a same scene. For 57 of these scenes, the main changes are photogrammetric and the remaining 59 show significant geometric deformations due to viewpoint changes on planar scenes.

Setup. ELF is tested on three classification networks trained on ImageNet, AlexNet [98], VGG [171], Xception [38] as well as the trained SuperPoint's and LF-Net's descriptor networks. Each variant is called after the network it relies on prefixed with ELF. The author's models are converted to Tensorflow [5] except for LF-Net. These variants provide

observations on the influence on the network architecture, the training task and training dataset.

Results. Figure 5.5 (left) shows that the *rep* variance is low across detectors whereas *ms* is more discriminative, hence our validation method (Section 4.1). On HPatches, SuperPoint (SP) reaches the best *rep-ms* [68.6, 57.1] closely followed by the ELF variants (e.g. ELF-VGG: [63.8, 51.8]) and TILDE [66.0, 46.7]. In general, learning-based methods all outperform hand-crafted ones. Still, LF-Net and LIFT curiously underperform on HPatches: one reason may be that the data they are trained on differs too much from this one. LIFT is trained on outdoor images only and two LF-Net models are available, one for indoor and one for outdoor images. However, HPatches holds both indoor and outdoor scenes. For fair comparison, LF-Net models are tested and the best results are reported, which is achieved by the indoor model. Even though LF-Net and LIFT fall behind the top learned methods, they still outperform classic hand-crafted methods, which suggests that these learned models embed more information than the hand-crafted methods. This supports the recent direction towards trained detectors and descriptors.

5.4.2 Illumination Robustness.

Dataset. The Webcam dataset [191] gathers static outdoor scenes with drastic natural light changes contrary to the HPatches images which mostly hold artificial light changes in indoor scenes.

Results. Once more, *ms* is a better discriminant on Webcam than *rep* (Figure 5.6 bottom). ELF-VGG reaches top *rep-ms* [53.2, 43.7] closely followed by TILDE [52.5, 34.7] which was the SoA detector.

Overall, there is a performance degradation (~20%) from HPatches to Webcam. The former holds images with standard features such as corners that state-of-the-art methods are made to recognize either by definition or by supervision. There are fewer features in the Webcam dataset because of the natural lighting that blurs them. There are also strong intensity variations that these models do not handle well. One reason may be that the learning-based methods never saw such lighting variations in their training set. However, this assumption is rejected since even SuperPoint, which is trained on Coco images, outperforms LIFT and LF-Net, which are trained on outdoor images. Another justification can be that what matters the most is the pixel distribution the network is trained on, rather than the image content. The top methods are the ELF variants

together with SuperPoint: the first ones are trained on the huge Imagenet dataset and benefit from heavy data augmentation. SuperPoint also employs a heavy synthetic data strategy to train its network. Thus, these CNNs may cover a much wider pixel distribution which would explain their robustness to pixel distribution changes such as light modifications.

5.4.3 Rotation and Scale Robustness

Dataset. Two synthetic datasets are derived from HPatches. For each of the 116 scenes, the first image is rotated with angles from 0° to 210° with an interval of 40°. Four zoomed-in version of the image are generated with scales $[1.25, 1.5, 1.75, 2]$. The two datasets are released with the code.

Results. ELF-VGG is compared to SoA detectors and their respective descriptors (Figure 5.7). Repeatability is mostly stable for all methods: SIFT and SuperPoint are the most invariant whereas ELF follows the same variations as LIFT and LF-Net. Once again, *ms* better assesses the detectors' performance: SuperPoint is the most robust to scale changes, followed by LIFT and SIFT. ELF and LF-Net lose 50% of their matching score with the increasing scale. It is surprising to observe that LIFT is more scale-robust than LF-Net when the latter's global performance is higher. A reasonable explanation is that LIFT detects keypoints at 21 scales of the same image whereas LF-Net only runs its detector CNN on 5 scales. Nonetheless, ELF outperforms LF-Net without manual multi-scale processing.

Even though *rep* shows little variations (Figure 5.8), all learned methods' *ms* crash while only SIFT survives the rotation changes. This suggests that the orientation normalization step in SIFT's detection is indeed relevant. LIFT and LF-Net integrateto an analog normalization: both learn the keypoint orientation with a CNN either from the image patch for LIFT, or from the deep patch features for LF-Net. The second approach appears to provide better results compared to LIFT. Not surprisingly, our proxy-descriptor is not rotation invariant as the convolutions that make the CNN are not. This also explains why SuperPoint also crashes similarly. These results suggest that the orientation learning step in LIFT and LF-Net is relevant but its robustness could be improved.

5.4.4 3D Viewpoint Robustness

Dataset. Three Strecha scenes [177] with increasing viewpoint changes are used: *Fountain, Castle entry, Herzjesu-P8*. The viewpoint changes proposed by HPatches are lim-

ited to planar scenes which does not reflect the complexity of 3D structures. Since the ground-truth depths are not available anymore, a pseudo scaleless ground-truth depth is recovered using SfM [160]. They are also made available in the released code.

Results. While SIFT shows a clear advantage of pure-rotation robustness, it displays similar degradation as other methods on realistic rotation-and-translation on 3D structures. Figure 5.9 shows that all methods degrade uniformly. One could assume that this small data sample is not representative enough to run such robustness analysis. However, these results rather suggest that all methods have the same robustness to 3D viewpoint changes. Even though the previous analyses allow ranking the different feature extraction methods, each has advantages over others on certain situations: ELF or SuperPoint on general homography matches, or SIFT on rotation robustness. This is why the experiments only aim at showing that ELF reaches the same performances and shares similar properties to existing methods as there are no generic ranking criteria.

5.4.5 Architecture influence

Rationale. The comparison of the classification network studies the influence of the representation space size and the convolution method: VGG and AlexNet use the same type of convolution but the functional space of VGG is much bigger than for AlexNet. VGG and Xception both have high dimension representation spaces but VGG uses classic convolution whereas Xception uses fusions of depth-wise convolutions. The comparison with SuperPoint and LF-Net aims at showing whether ELF can benefit from a network trained for feature extraction.

Results. ELF is applied to three classification networks and the descriptor networks of SuperPoint and LF-Net (Figure 5.5, 5.6 - 'Our variants').

For a fixed training task (classification) on a fixed dataset (ImageNet), the VGG, the AlexNet, and the Xception variants are compared. As could be expected, the network architecture has a critical impact on the detection and ELF-VGG outperforms the other variants. One explanation is that AlexNet is made of wider convolutions than VGG, which induces a higher loss of resolution when computing the gradient. As for *ms*, the higher representation space of VGG may help to build more informative features which are a stronger signal to backpropagate. This could also justify why ELF-VGG outperforms ELF-Xception that has fewer parameters. Another explanation is that ELF-Xception's gradient maps seem smoother. Salient locations are then less emphasized

which makes the keypoint detection harder. One could hint at the depth-wise convolution to explain this visual aspect but no experimental way to verify it was found for now. Surprisingly, ELF-LF-Net outperforms the original LF-Net on both HPatches and Webcam and ELF-SuperPoint variant reaches similar results as the original.

5.4.6 Individual components comparison

Setup. A first comparison experiment evaluates how well the individual detector performs in feature extraction with our proxy descriptor. This provides a comparison between salient points that a CNN autonomously learns and the ones defined by hand or by supervision. It also raises the question of whether contrastive and triplet losses are relevant for descriptor training compared to the description a CNN naturally learns.

A second comparison measures how well the ELF detector can integrate into other pipelines by replacing their original detector with ours. This brings attention to whether detector training aims at learning information already embedded in networks.

Results. First, the descriptor of all methods is replaced with the ELF-VGG descriptor using the $pool_3$ layer. They are then compared to the top ELF variant based on VGG (Figure 5.10, strips). Here, $pool_3$ is selected instead of $pool_4$ because it produces better results for the other methods while preserving ours. ELF reaches higher *ms* [51.3] for all methods except for SuperPoint [53.7] for which it is comparable. This shows that it is as relevant, if not more, than previous hand-crafted or learned ones.

This naturally leads to the question: *'What kind of keypoints does ELF detect ?'*. There is currently no answer to this question as it is complex to explicitly characterize properties of the pixel areas around keypoints. Hence the open question *'What makes a good keypoints ?'* that the research still tries to answer. Empirically, the ELF detector activates mostly on high-intensity gradient areas although not all of them. An assumption is that, as the CNN is trained on the vision task, it learns to ignore image regions useless for its semantic representation. This results in killing the gradient signals in those areas that may be unsuited for matching.

Another surprising observation is that for a fixed SuperPoint (SP) detector, SP descriptor and our proxy-descriptor reach similar *ms*. This raises the question of whether contrastive-like losses can better constrain the CNN features than simpler losses. This also shows that there is more to CNNs than only the task they are trained on: they embed much more information that can prove useful for unrelated tasks. Although the simple proxy-descriptor is defined for evaluation purposes, these results demonstrate that it can be used as a description baseline for feature extraction.

The integration of the ELF detector with other method's descriptor (Figure 5.10, circle) also boosts the *ms*. [199] previously suggested that there may be a correlation between the detector and the descriptor when they are jointly trained, *i.e.* the LIFT descriptor is trained to describe only the keypoints output by its detector. However, these results show that ELF can easily be integrated into existing pipelines and even boost their performances.

5.4.7 Gradient Baseline

Setup. Visually, the feature gradient map is reminiscent of the image gradients computed with the Sobel or Laplacian operators. To evaluate the difference, two ELF variants are run where the saliency map is replaced with the image standard gradient. This aims at showing whether feature gradients embed more information than image intensity variations only.

Results. The saliency map is replaced with simple Sobel and Laplacian gradient maps of which local maxima are keypoints. The repeatability of these points is plotted Figure 5.11 - Left. These two gradients are completed with the descriptors from ELF on VGG, AlexNet, and Xception. The matching performances are compared with their respective ELF variant (Right). Results show that these simpler gradients can detect systematic keypoints with comparable *rep* on very structured images such as HPatches. However, ELF is more robust to illumination changes (Webcam). On HPatches, the Laplacian-variant reaches similar *ms* as ELF-VGG (55 *vs* 56) and outperforms ELF-AlexNet and ELF-Xception. One explanation is that when the images are structured, high-intensity gradient locations are relevant enough keypoints. However, on Webcam, all the ELF detectors outperform the Laplacian and Sobel gradients with a factor of 100%. This shows that ELF is more robust than the Laplacian and Sobel operators. Also, feature gradient is a sparse signal which is better suited for local maxima detection than the much smoother Laplacian operator (Figure 5.12).

Qualitative results The green lines Fig. 5.13 are ELF's putative matches based only on nearest-neighbor matching of descriptors. More qualitative results are presented in the video [3].

[3] https://youtu.be/oxbG5162yDs

5.5 Summary

This chapter introduced ELF, a novel method to extract feature locations from pre-trained CNNs, with no further training. Extensive experiments show that it performs as well as SoA detectors. It can easily be integrated into existing feature extraction pipelines and proves to boost their matching performances. Even when completed with a simple feature-map-based descriptor, it turns into a competitive feature extraction method. These results shed new light on the information embedded inside trained CNNs. This work also raises questions on the descriptor training of deep-learning approaches and whether their losses constrain the CNN to learn better features than the ones it would learn on its own to complete a visual task. Preliminary results show that the CNN architecture, the training task, and the dataset have a consequent impact on the detector performances. Further analysis of these correlations is the object of a future work.

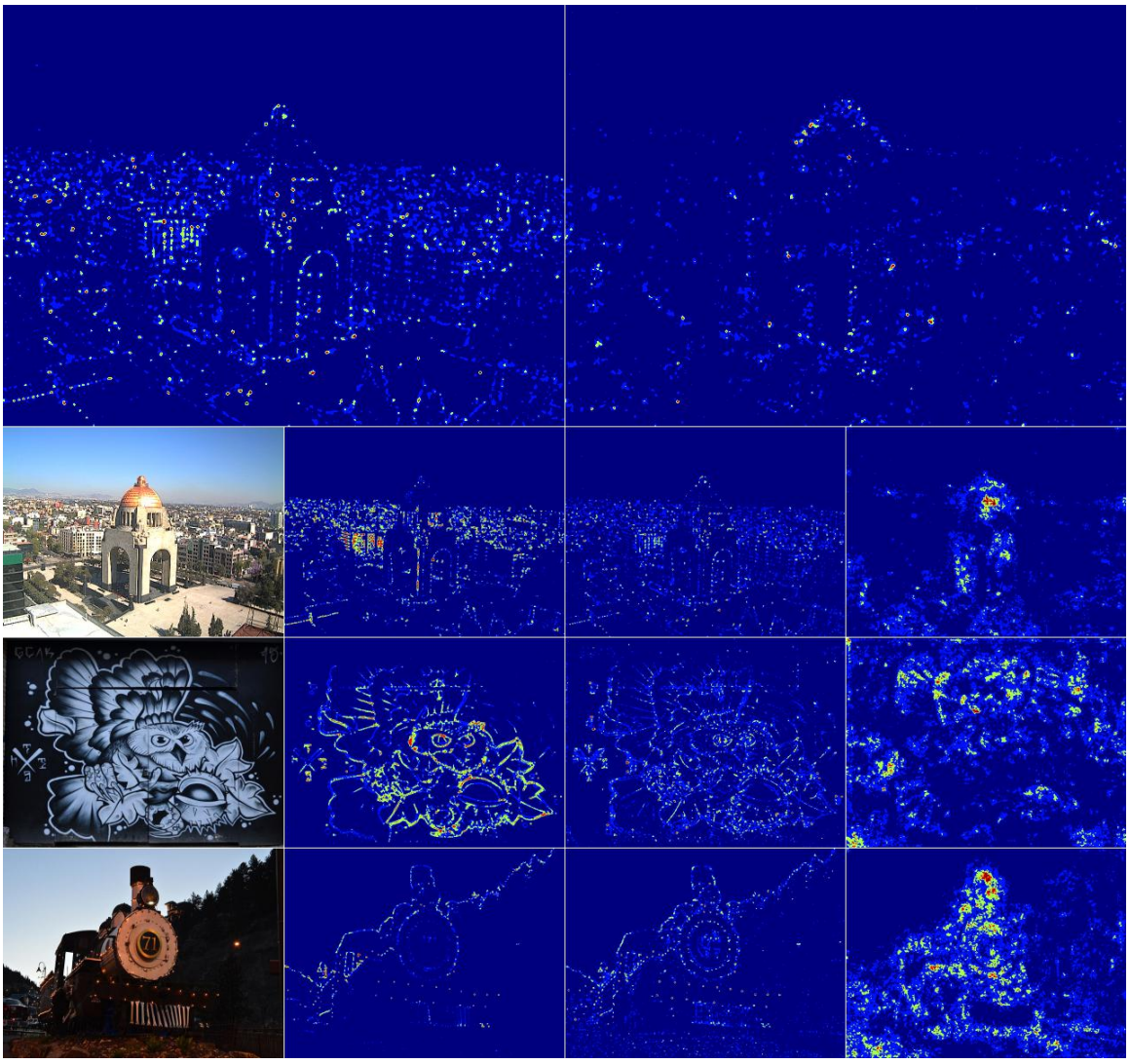


Figure 5.2: Saliency maps computed from the feature map gradient $S^l(\mathbf{I}) = \left|\nabla_I F^l(I)^2\right|$. Enhanced image contrast for better visualization. Top row: gradients of VGG $pool_2$ and $pool_3$ show a loss of resolution from $pool_2$ to $pool_3$. Bottom: $(pool_i)_{i\in[1,2,5]}$ of VGG on Webcam, HPatches and Coco images. Low-level saliency maps activate accurately whereas higher saliency maps are blurred.

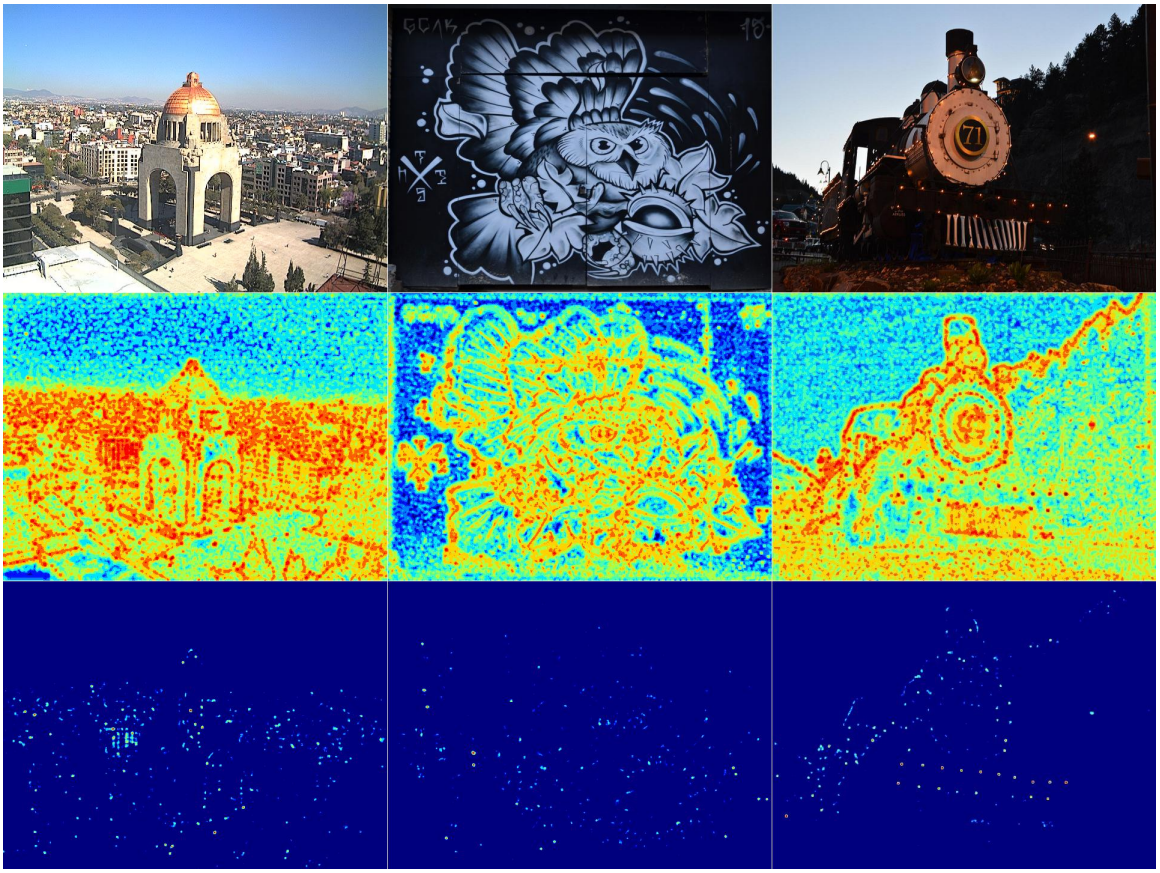


Figure 5.3: Saliency maps thresholding to keep only the most informative location. Top: original image. Middle: blurred saliency maps. Bottom: saliency map after thresholding.

Figure 5.4: Preview of the evaluation datasets. Left-Right: HPatches: planar viewpoint. Webcam: light. HPatches: rotation. HPatches: scale. Strecha: 3D viewpoint.

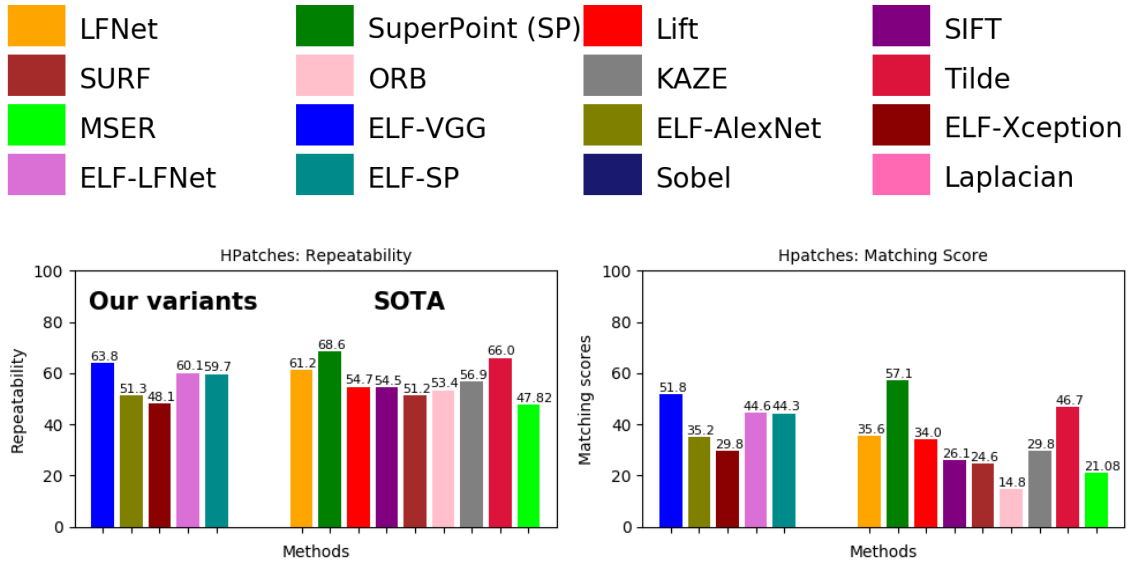


Figure 5.5: Local feature matching performance on HPatches [19]. Left-Right: repeatability, matching score.

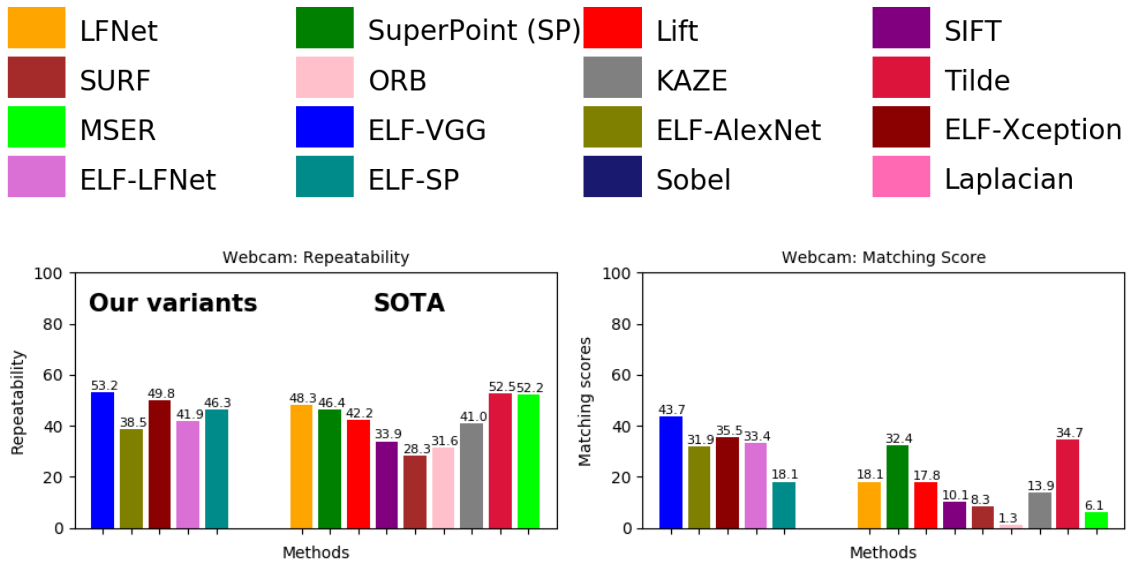


Figure 5.6: Local feature matching performance on Webcam [19]. Left-Right: repeatability, matching score.

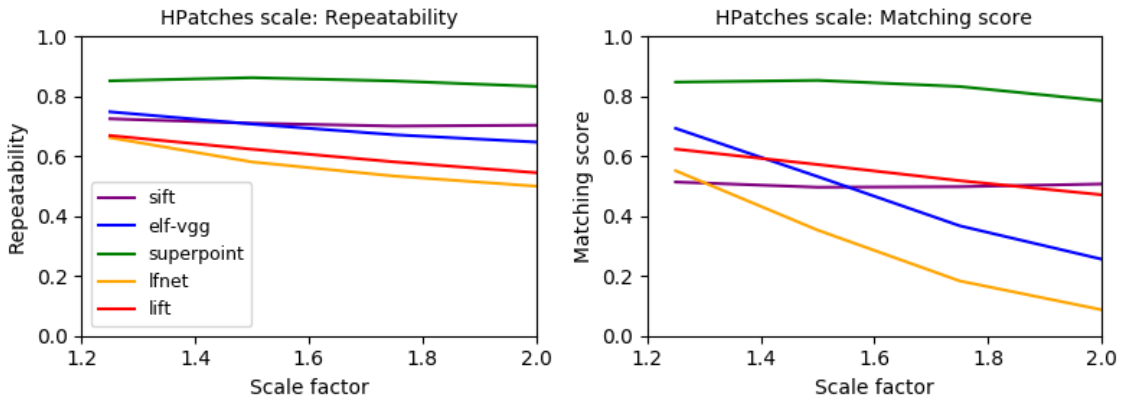


Figure 5.7: HPatches scale robustness. Left-Right: rep, ms.

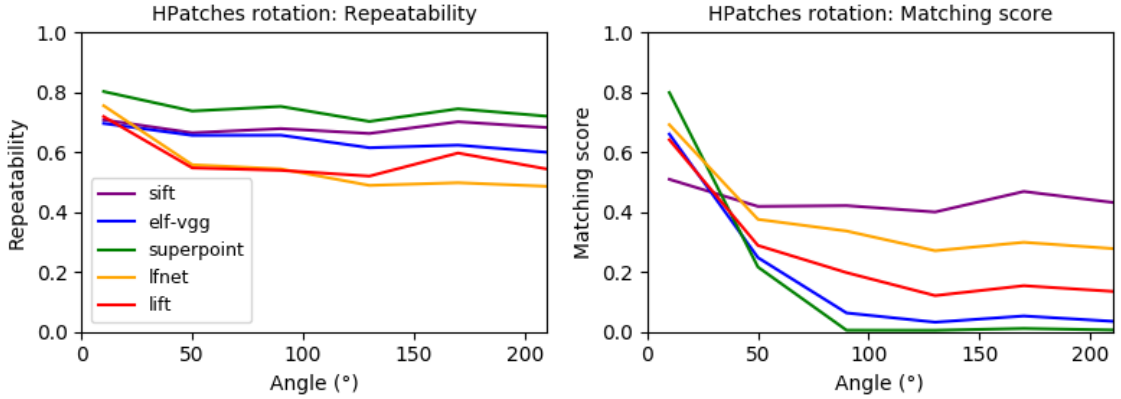


Figure 5.8: HPatches rotation robustness. Left-Right: rep, ms.

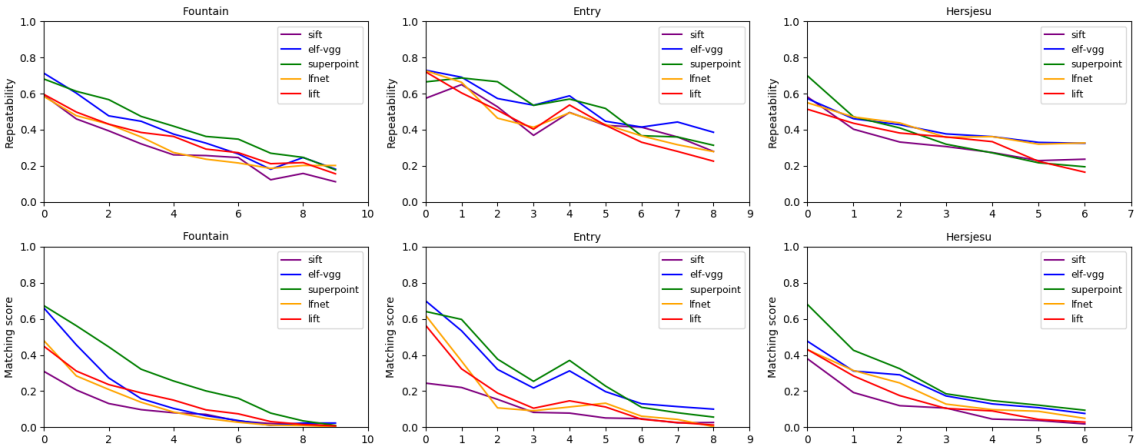


Figure 5.9: Robustness analysis: 3D viewpoint.

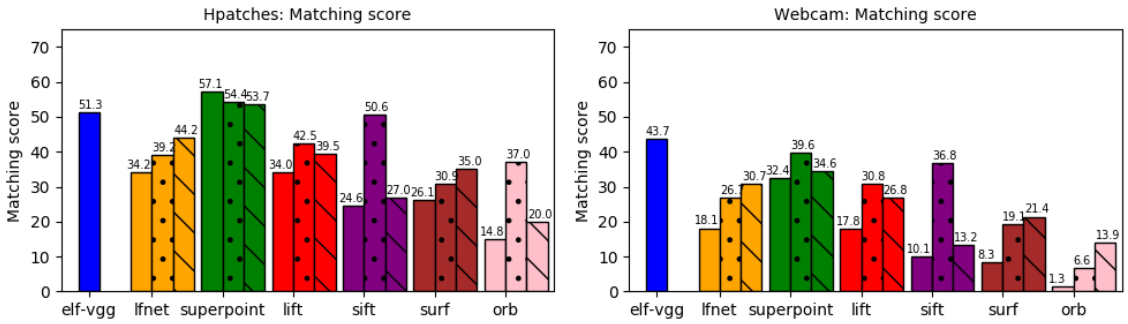


Figure 5.10: Left-Right: original perf, integration of ELF, integration of the VGG-proxy-descriptor.

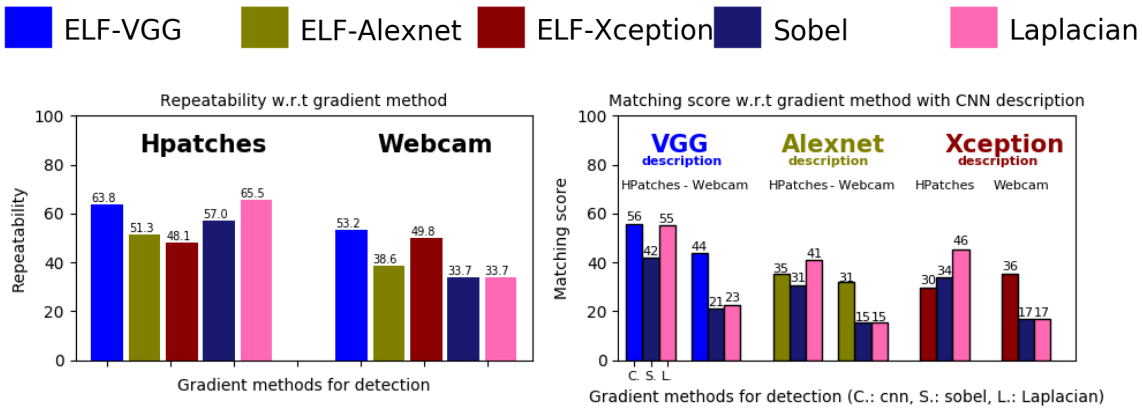


Figure 5.11: Gradient baseline.

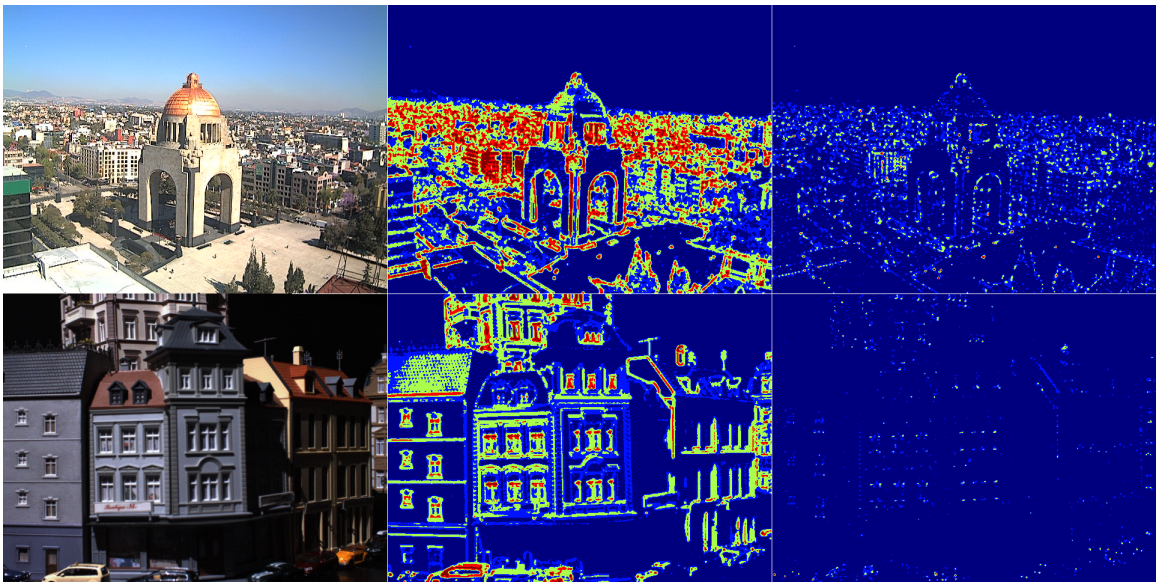


Figure 5.12: Feature gradient (right) provides a sparser signal than Laplacian (middle) which is more selective of salient areas.

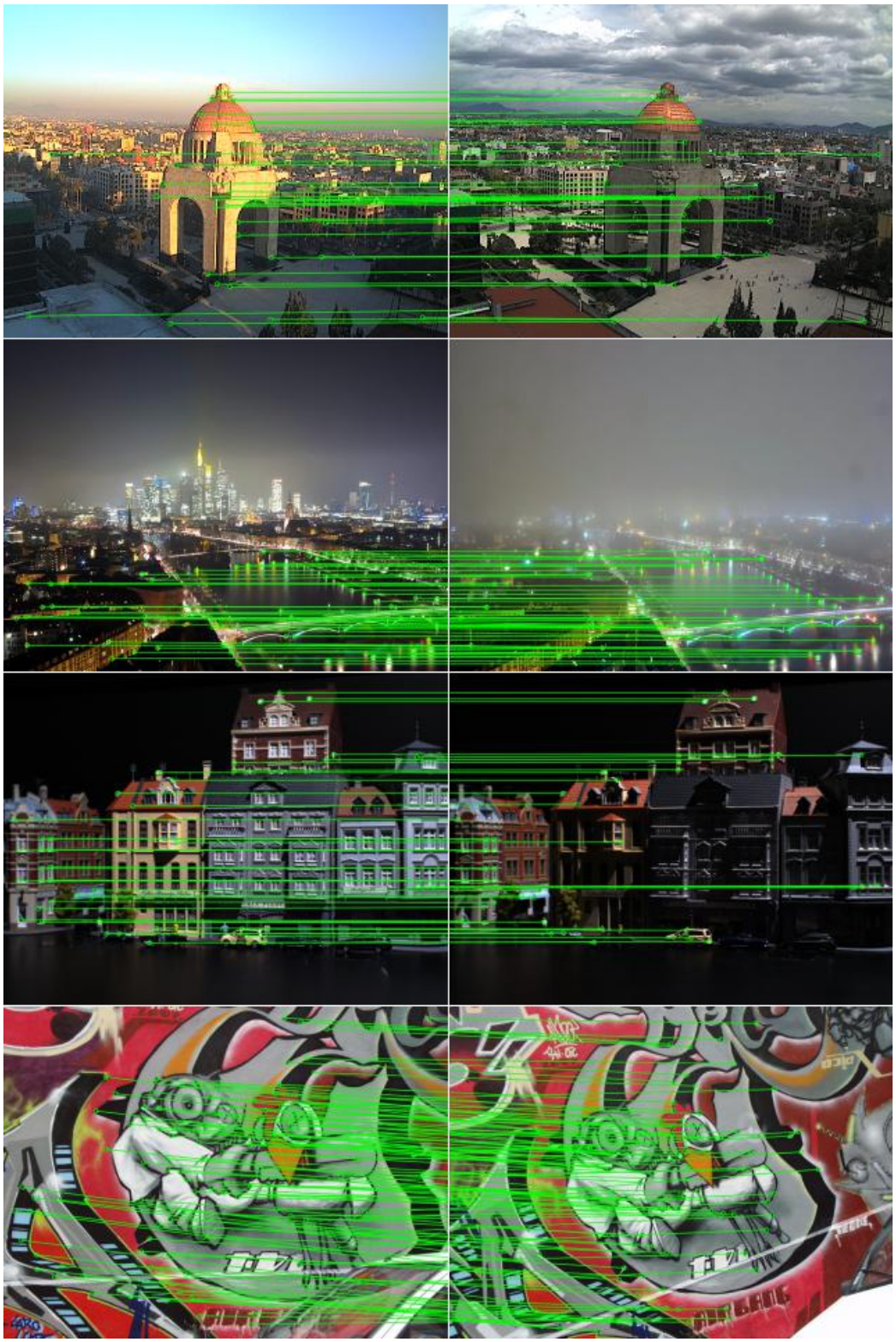


Figure 5.13: Green lines are ELF's putative matches of the proxy-descriptor before RANSAC-based homography estimation.

Part III

Semantics for Robust Localisation

The previous contributions of this thesis related to visual features. The first part addressed the problem of feature robustness for long-term scene recognition. It answered this problem with two novel image descriptors based on semantics and geometry. The second part proposed to reduce the need for heavy training to get data-specific local features. It built on previous work that already suggests that the representation space of a CNN can answer this problem.

The remaining chapters of this thesis focus on the better exploitation of semantic information. The global descriptors introduced previously already showed that semantics can be integrated with other image information to provide robust solutions for visual tasks. This part describes another example where the integration of semantic into existing edge-based visual odometry algorithms improves their tracking robustness.

Chapter 6

Integration of Semantic Edge for Direct Visual Odometry

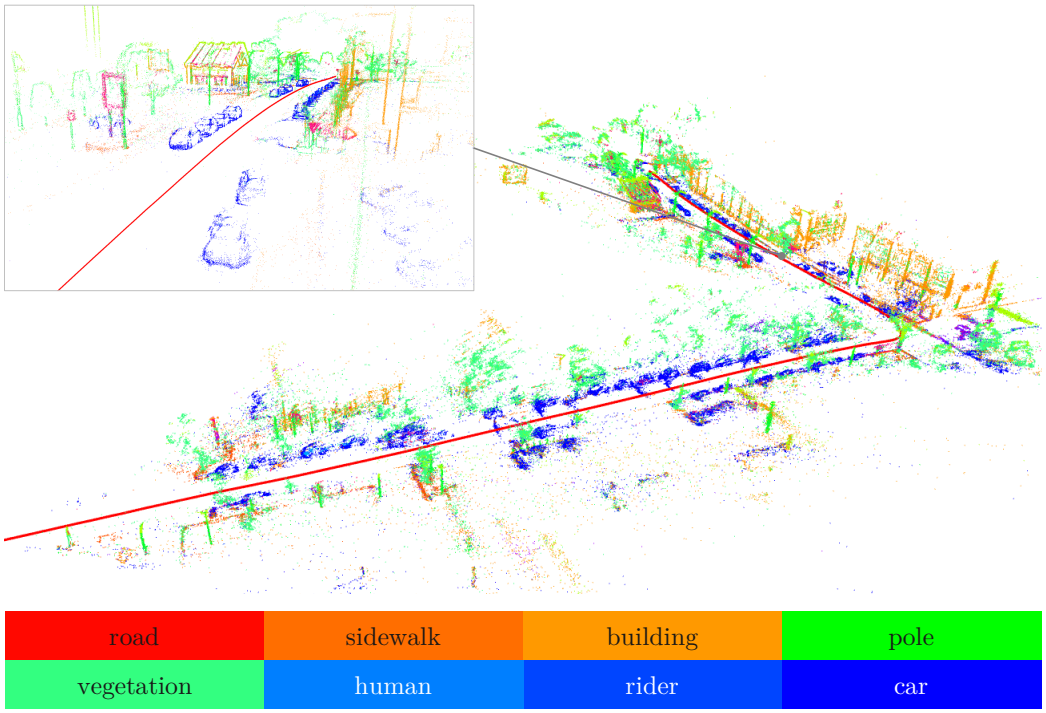


Figure 6.1: Visualisation of the semantic VO output. Red line: recovered trajectory. Color point cloud: the reconstructed scene where the color of 3D point represents its semantic class.

The contributions presented in this chapter are the results of a collaboration with my Ph.D. colleague Xiaolong Wu, whose main research area is robust outdoor visual odometry. In this joint work, he contributed to the optimization derivation and implementation, and I contributed to the data collection and generation. This work resulted in a paper that we jointly wrote and submitted to IROS19 [195]. This chapter borrows

the plots from the paper but the writing is my own.

The first two chapters of this thesis showed that the integration of semantics into visual local features [23, 25, 182, 201] is relevant to make scene recognition robust to strong variations in appearance. This chapter investigates whether semantics can also prove useful for robust outdoor direct visual odometry.

Sec. 2.1 previously introduced the direct VO optimization and its challenges. Given two images of the same scene taken under different viewpoints, it iteratively optimizes the camera displacement and the depth of the scene that corresponding pixels are projected to the same 3D point in the scene. In practice, once the depth and the camera pose are estimated, the first image is warped over the second and the optimization is evaluated with the image alignment. The convergence and the performance of this algorithm heavily depend on this alignment loss. A standard way to derive it is to compare the pixel intensities between the warped and target images, which assumes that the brightness is constant between the two images. This assumption is easily violated in outdoor environments. Another loss evaluates only the alignment of the edges instead of the whole image. Edges are more robust to light variations that pixel intensity but they usually are not repeatable enough from one image to another. This means that one edge in one image may not appear in the other one so it can not be aligned and contribute to the optimization loss. Even worse, it can be wrongly aligned with another edge of similar geometry and lead the VO to diverge. This can happen when the camera's displacement is too large and the resulting viewpoint differ too much. This motivates this chapter to investigate the integration of semantic edges in monocular direct VO and their influence of the optimization's robustness. Experiments show that semantic edges are more repeatable than standard ones, which make the algorithm more robust to viewpoint variations. Also, semantics provide additional constraints on the pixels to align so the final camera pose is more accurate.

The rest of this chapter is organized as follows. Sec. 6.1 recalls the edge-based VO derivation. Sec. 6.2 derives the integration of semantics into the VO optimization. The method is dubbed SNNFs for Semantic Nearest Neighbor Fields. Sec. 6.3 presents the accuracy and robustness performance of this novel optimization in an autonomous driving setting on the KITTI dataset. Results show that this system achieves SoA performances.

6.1 Review of Direct Visual Odometry

This section briefly recalls the visual odometry principles previously described in Sec. 2.1.

Visual Odometry (VO) is the process by which a system mounted with a camera relies only on images to recover its trajectory inside an estimated map of the world Fig. 6.1. It is widely used in many outdoor robotic applications such as autonomous driving or environmental monitoring. Existing methods usually fall in one of these two categories: direct or indirect methods.

Indirect methods [94, 131, 176] rely on local feature matching between overlapping images to recover the camera displacement. These methods rely on the research efforts on robust local features to handle both photometric noise and geometric distortion in images.

Direct methods jointly optimize the camera displacement and the scene depth to align successive images. These estimations are used to warp the first image onto the second, and the estimation is evaluated by the pixel intensity difference between the warped and the second images. But this makes the unreasonable assumption that the brightness stays constant between successive images. This limits the use of direct VO outdoor where the light is out of the system's control. Recent contributions [53, 52, 132, 138, 144] address this problem and improve the accuracy of the motion estimation. However, another problem is the small convergence basin of direct optimization compared to indirect methods because of the unconstrained data association: local feature matching is constrained by the descriptor distance. Edge-based VO estimation is a relevant alternative to address this issue.

In a way, edge VO reduces the gap between direct and indirect methods: it matches local edges between two images by recovering the camera motion and the depth from the geometric constraint of the matches. The difference with the feature-based approaches is that edges are associated using Iterative Closest Point (ICP)-based alignment rather than using descriptor matching. Since edges are more robust than pixels against image variations (*e.g.*, illumination changes, motion blur, or occlusion), they are more reliable to align. This allows edge-based motion estimation to reach impressive motion accuracy in indoor environments [213]. Yet, when it comes to outdoor environments, the poor repeatability of standard edges detectors break the performance. This is because the existing VO methods rely on a simple edge-association strategy that is highly sensitive to outlier edges and large camera motion. This motivates this chapter to investigate a robust edge association strategy using semantics invariants. Another source of robustness is the recent effort in edge learning towards the detection of useful edges only. One advantage of these new approaches is that they filter out most of the noisy edges that are not exploitable by the VO. This chapter also investigates the advantages of these learned edges over the standard Canny detector [33].

6.2 Semantic-Edge Alignment-Based optimization

The derivation of edge-based VO is recalled before introducing the semantic adaptation.

Notations. The Euclidean coordinates of a point $\boldsymbol{X}$ in the 3D scene are $\boldsymbol{X} = (x, y, z)^{\mathrm{T}}$ in the camera frame. The camera coordinate systems is the one typically used in computer vision. The camera looks down the z-axis, with the x-axis pointing to the right and the y-axis pointing downwards. The z coordinate is the depth of the point. The 3D rotation and translation that transforms coordinates from frame c to c' are written ${}^{c'}\mathrm{R}_c \in \mathrm{SO}(3)$ and ${}^{c'}t_c \in \mathbb{R}^3$, with SO(3) denoting the group of orthogonal matrices of size 3.

This chapter works on gray-scale images $\mathrm{I} : \mathbb{R}^2 \to \mathbb{R}$ where $\boldsymbol{x}$ is the 2D pixel coordinates and $\bar{\boldsymbol{x}}$ denotes the homogeneous coordinates (See Sec. 2.1 for more details). These two coordinates are related by Eq. (6.1):

$$\boldsymbol{X} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad \bar{\boldsymbol{x}} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \to \boldsymbol{x} = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix} \tag{6.1}$$

The $\mathscr{L}_2$ loss is written $\|\cdot\|_2$ and the Huber loss $\|\cdot\|_\gamma$. The camera intrinsic matrix is noted $\mathrm{K} \in \mathbb{R}^{3\times3}$. The set of edge pixels in one image is noted $\mathscr{E}$.

6.2.1 Edge-Based Optimization

Edge-based VO estimates the camera motion between a reference camera frame and a target frame. Edges are detected in the reference image I_r and the new image I_k. Given the camera motion between the two frames and the depth of the scene, the edge pixels in I_r are projected onto I_k. Each projected pixel is assigned to the nearest edge pixel detected in I_k. Given ideal edges, the optimization finds the camera motion and the depth that minimizes the distance between these associated pixels. This is why repeatable edges are pivotal for this optimization. When they are not, there is no way to know if the pixel matches are incorrect because the estimation is false or because they belong to different edges.

The mathematical derivation in the rest of this paragraph formalizes the previous paragraph. It is not required to understand the rest of this chapter.

Let $\boldsymbol{x} = (u, v) \in \mathscr{E}$ be an edge pixel in the source image (camera frame c) with depth z. It is projected onto a pixel $\pi(\boldsymbol{x})$ in the target image (camera frame c') using the operation:

$$\pi(\boldsymbol{x}) = \begin{pmatrix} \frac{\tilde{u}}{\tilde{w}} \\ \frac{\tilde{v}}{\tilde{w}} \end{pmatrix} with \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \mathrm{K} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} {}^{c'}\mathrm{R}_c & {}^{c'}t_c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z \cdot \mathrm{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}_c \\ 1 \end{pmatrix} \tag{6.2}$$

Note that the unknown in Eq. (6.2) are the pixel depth z and the camera rotation ${}^{c'}\mathrm{R}_c$ and the translation ${}^{c'}t_c$.

Each projected pixel $\pi(\boldsymbol{x})$ is associated to the nearest pixel in $\mathscr{E}'$, the set of edge pixels detected in the target image I':

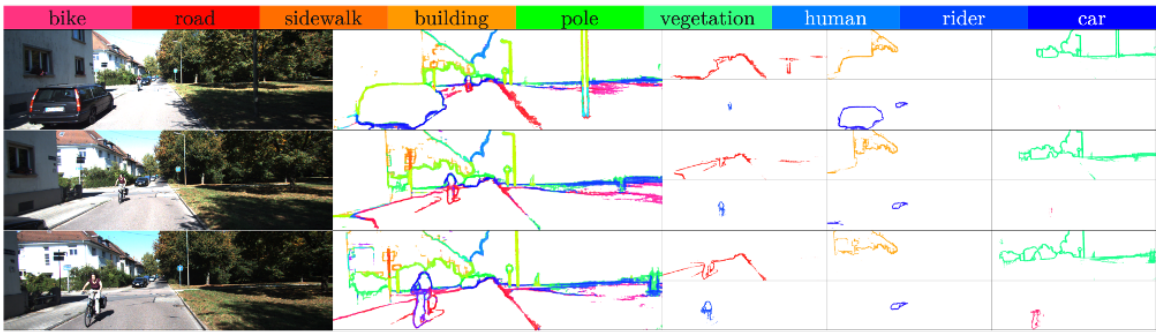
$$\mathrm{NN}(\pi(\boldsymbol{x})) = \underset{\boldsymbol{x}' \in \mathscr{E}'}{\operatorname{argmin}} \| \boldsymbol{x}' - \pi(\boldsymbol{x}) \| \tag{6.3}$$

The optimization estimates the camera rotation and translation, and the pixel depth, that minimizes the distance between associated edge pixels:

$$\mathrm{E} := \sum_{\boldsymbol{x} \in \mathscr{E}} \| \pi(\boldsymbol{x}) - \mathrm{NN}[\pi(\boldsymbol{x})] \|_\gamma \tag{6.4}$$

The energy function in Eq. (6.4) is minimized using a 2D-3D ICP-based optimization [95]. It alternates between finding approximate nearest neighbors and register the putative correspondences using an iteratively reweighted Gauss-Newton algorithm. Following the theory of optimization under unitary constraints [119], the energy function is minimized on Lie-manifolds for better convergence.

6.2.2 Semantic Nearest Neighbor Fields



(a) RGB image (b) semantic edges (c) semantic edges in SNNF

Figure 6.2: Illustration on the semantic edge extraction on the KITTI dataset. The first and second columns show the image and all its semantic edges. The rest of the columns show a subset of the semantic edges.

The previous optimization is now adapted to integrate semantic constraints at the pixel association step 6.3. Previously, the projected edge pixel was assigned to the nearest edge pixel in the target image. Now, the associated pixel must also have the same semantic class as the projected pixel.

The semantic edges are generated with available CNN models, such as CaseNet [202]. As described in Sec. 3.3, these networks output probability maps at the image resolution that represents the probability of a pixel to belong to a semantic edge of a given class. More formally, given an image $\mathrm{I} \in \mathbb{N}^{h \times w \times 3}$ of height $h \in \mathbb{N}$ and width $w \in \mathbb{N}$, K semantic labels, the network outputs K pseudo-probability maps $(\mathrm{Y}_k)_{k \in [\![0,\mathrm{K}-1]\!]}$, with $\mathrm{Y}_k \in [-1,1]^{h \times w}$. Each map Y_k represents the probability of the pixels to belong to a semantic class k, *i.e.*, $\mathrm{Y}_k(u,v)$ is the probability that pixel (u,v) belongs to the semantic edge of class k. Note that the CaseNet network is designed such that $\sum_{k=0}^{\mathrm{K}-1} \mathrm{Y}_k(u,v) \neq 1$, *i.e*, a pixel can belong to several classes with probabilities higher than 0.5 for each. For example, an edge pixel lying between a car and the road has both labels. The labels CaseNet is trained on are the Cityscape labels described in Tab. 6.1.

road	pole	terrain	truck	car
sidewalk	traffic light	sky	bus	fence
building	traffic sign	person	train	
wall	vegetation	rider	motorcycle	

Table 6.1: Cityscapes labels.

During the data association step, each edge pixel is only matched with pixels that have the same semantic label. This reduces ambiguous associations and enlarges the convergence basin. Since a pixel has multiple classes, it is associated several times, one for each of its classes. By adding these constraints, this soft data association makes the optimization more robust. The energy function 6.4 is adapted to integrate these additional constraints. The set of edge pixels with class k in the reference image is now written $\mathscr{E}_k$. The search for the nearest edge pixel in the target image with class k written NN_k. The new energy function is defined by:

$$\mathrm{E} := \sum_{k=1}^{\mathrm{K}-1} \sum_{\boldsymbol{x} \in \mathscr{E}_k} \| \pi(\boldsymbol{x}) - \mathrm{NN}_k[\pi(\boldsymbol{x})] \|_\gamma \tag{6.5}$$

6.3 Experiments

Datasets. The localization is evaluated on the left-camera images from the KITTI odometry dataset [62]. The experiments are run on the rectified color images for which intrinsic and extrinsic are provided. The data is split into 11 sequences (00-10) and each of them exhibits specific semantic content summarised in Table 6.2. This allows assessing the influence of the semantics on the odometry performance.

Scene	Sequence No.	Semantics
city	00, 05, 06, 07	buildings, cars with few vegetation
village	02, 03, 04, 08, 09	vegetation with few buildings, cars
highway	01	roads, cars, and signs

Table 6.2: Semantics in various KITTI dataset sequences.

Baselines. The experiments compare SNNFs to its non-semantic counterparts ANNFs [213] and ONNFs [214] to evaluate the robustness gain induced by the semantic constraints. These methods are tested with the default Canny detector and the learned edges Holistically-nested edge detection (HED) [198] and Structured Edges (SE) [47]. They are also evaluated with the semantic edges from CaseNet [202] and SEAL [203]. The probabilistic edges are fused over the semantic labels to output one semantic-less edge map. This evaluates the influence of edge detection on these methods' performance.

Setup. The semantic edges are generated with the CaseNet and SEAL variants trained on the Cityscapes dataset [**?**]. The learned edges models for SE [47] and HED [198] used in this chapter are the ones trained on the BSDS500 dataset [12, 120]. The learned edges are generated with the code released by the authors. SE is run with Matlab 2017 and HED, CaseNet and SEAL rely on the Caffe [85] on an Nvidia 1080Ti set with Cuda8 and CudNN6. The semantic edges are generated with the CaseNet and SEAL variants trained on the Cityscapes dataset [**?**].

Following [213], we implement a point-to-tangent residual, that is we project the original pixel-wise residual onto its local gradient direction to obtain additional robustness against outliers. It should be noted that this formulation makes the underlying assumption that the camera motion is free of large inter-frame rotations. In reality, this assumption is valid for the autonomous driving application considered in this chapter.

The semantic edge constraints are integrated both into mapping and tracking. In the tracking phase, the edge residuals get more weights to enforce a better convergence basin. In the mapping phase, they get fewer weights and the depth map is regularized to

penalize large inverse depth updates. For example, the inverse depth of an edge pixel is unobservable when the epipolar lines are perpendicular to the edge normals.

In some experiments, the edge-based VO is integrated into the standard pixel-based VO with the difference that the pixel intensity is replaced with the intensity gradient. The gradient proves to be more robust to illumination changes [196].

When the images exhibit few edges, the pixels that support the optimization are not distributed uniformly over the image. This often happens when the scene is dominated by vegetation and can lead to ambiguous motion estimation. To solve this problem, additional supportive pixels are sampled even when they are non-edge pixels. When the edges are well distributed, only a few pixels are needed and vice-versa. The sampling strategy in the same as in [52]. These points are not integrated into the semantic data-association. Instead, they are registered only based on their photometric gradient similarity as in standard direct VO.

Metrics. The localization is evaluated with the Absolute Trajectory Error (ATE). It measures the absolute difference between the camera positions of two trajectories. For a fair comparison, no loop-closure is used and all methods use the ground truth poses to recover the scale of motion every 200 frames. Experiments show that the pose estimation for the first frames are usually unstable and vary for each method. So the first ten pose estimates are discarded for all runs.

6.3.1 Localization Performances

SNNF is compared to the following SoA methods: mono-ORBSLAM2 [131] for monocular indirect VO, DSO [52] for direct approaches and VSO [109] for the semantic direct methods. Overall, SNNF reaches lower error in the camera pose estimation for all urban scenes. It also exhibits a larger convergence basin which makes it more robust to large camera displacement than other edge-based approaches. However, it is limited by the segmentation performance: when the edges are too noisy, the performance can drop but still achieves SoA. The rest of this section details the evaluation and Fig. 6.3 shows examples of recovered trajectories.

Setup. The authors of both ORB-SLAM2 and DSO provide code to run the experiments. There is no code release with the paper describing VSO. So an alternative implementation is developed by introducing the semantic constraint energy into DSO for both tracking and mapping. For fair comparison, only 4000 active points are kept in all methods. Here, all edge-based methods are augmented with the gradient photometric constraint.

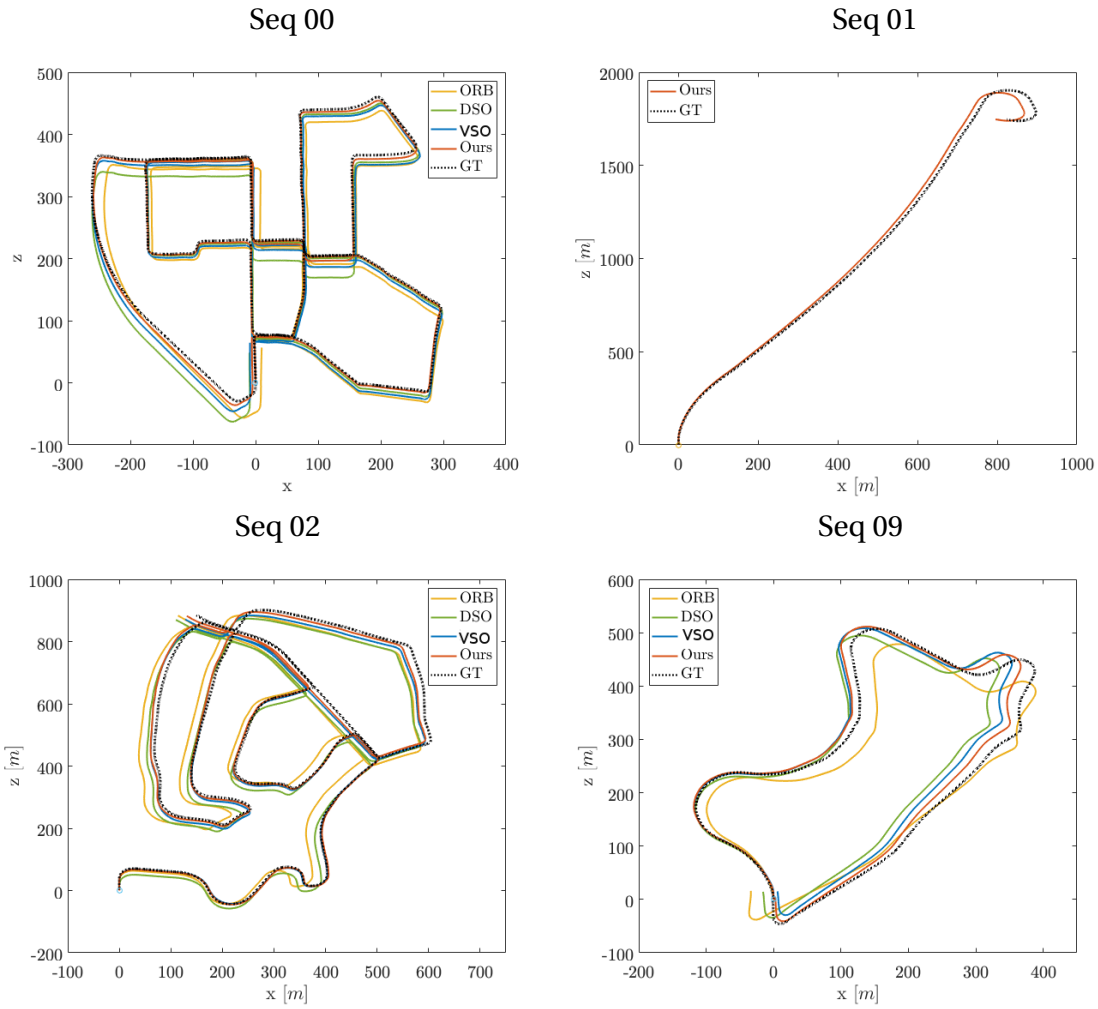


Figure 6.3: Qualitative results: Trajectories recovered from SNNF, indirect ORBSLAM2, direct DSO, and semantic VSO systems on KITTI. Left to Right: KITTI-seq00, 01, 02, and 09. Note that seq01 only shows SNNF and the ground truth because other methods cannot generate the whole trajectory.

Results. Table 6.3 summarises the ATE of the camera for all KITTI sequences. SNNF provides lower error than the SoA, especially on the highway sequence. For this sequence, only SNNF converges whereas VSO and DSO can not recover the full trajectory. One reason is the nature of the scene and the movement. The sequence holds mostly images from the highway road with either trees along the road or grasslands. This means that these images hold few discriminant features that ORB relies on, which explains the error higher than average for ORBSLAM2. It also holds poor texture that DSO relies on, so there are less discriminative regions to align during the VO optimization. This makes it easy for the DSO tracking to fails. And although VSO relies on semantics to garner more information, the scenes hold few large connected components that are harder to

KITTI	city				village					highway
	00	05	06	07	02	03	04	08	09	01
ORBSLAM2	16.14	15.96	13.35	10.63	15.58	**3.44**	3.05	15.43	12.88	36.32
DSO	16.83	13.64	16.83	9.55	17.08	3.71	**3.01**	18.31	13.05	-
VSO	15.31	10.08	14.10	8.39	14.57	3.76	3.09	15.29	13.12	-
SNNF	**11.82**	**8.39**	**10.92**	**6.11**	**14.15**	3.72	3.03	**15.07**	**12.63**	**14.59**

Table 6.3: Tracking Error on the KITTI sequences for SNNF and the SoA.

precisely align than in urban scenes with smaller semantic units and better semantic borders. This suggests that the geometric information inside the edges can boost the robustness of the VO, as supported by these numerical results.

In the city, SNNF reaches lower pose error than the SoA by a margin of up to 30% (seq 05). One explanation is that the segmentation is particularly accurate in the cities and exhibits many semantic edges due to the rich urban structures. This way, there are more reliable edges to exploit for the optimization. This also benefits VSO, which supports the integration of semantics benefits the VO.

This stays true even when the segmentation gets noisy as in the village sequences 03 and 04. For these trajectories, SNNF is outperformed by ORBSLAM2 and DSO by no more than 3%. One reason is that these sequences display more vegetation elements for which the semantic edges are less repeatable. This hinders the convergence to the correct pose.

6.3.2 Convergence Analysis

Rationale. One challenge of VO is the tracking robustness. One way to evaluate it is to measure how well the optimization can recover from large camera displacement. Experiments show that SNNF offers higher robustness than Approximate Nearest Neighbour Field (ANNF) and Orientation Nearest Neighbour Field (ONNF).

The tracking robustness is measured with the trajectory ATE with respect to the initial camera displacement. The displacement ranges from 0 to 5 meters since there is usually too little overlap for larger shifts.

Setup. To rule out the error introduced by the depth estimation's inaccuracies, the tests are run on the vKITTI dataset [58] instead of the KITTI one. The ground-truth depth is integrated into the optimization and only the camera motion is estimated. Since it is a simulation dataset, the depth is accurate enough to enable the tracking evaluation without the mapping noise. The tracking starts from the ground-truth camera pose at

the beginning of the trajectory. The next frame is sampled within a range of 5 meters. The rest of the optimization is the same as before. The ATE with respect to one value of initial displacement is averaged over all trajectories.

Here, the photogrammetric loss is not used in the optimization, and SNNF and ANNF are derived with the learned HED edges.

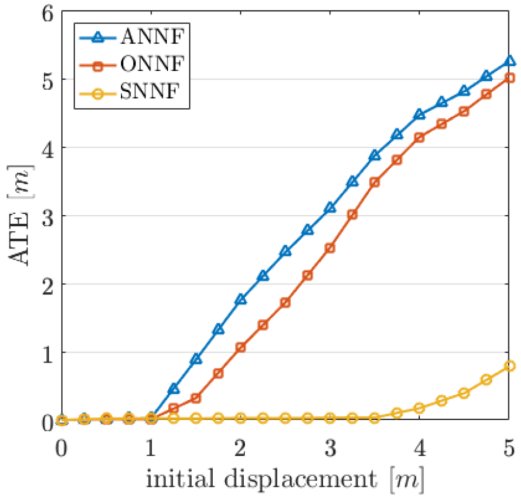


Figure 6.4: ATE averaged over vKITTI trajectories with respect to initial camera displacement. SNNF can recover from larger displacements than SoA edge-based approaches.

Results. Figure 6.4 shows that SNNF is more robust than edge-based approaches. The latter converge to trajectories which ATE becomes linear with the initial displacement once it is higher than 1 meter. This phenomenon occurs only after 3.5 meters for SNNF and the final error is sub-linear. Given that tracking failure is one of the main challenges tackled by VO research, such robustness is pivotal.

6.3.3 Edge Repeatability

Learned edges and more specifically semantic edges tend to be more repeatable than the standard Canny approach.

Rationale. The choice of edge detection for outdoor VO is still an open question. Schenk and Fraundorfer [159] observe that the performance of edge-based VO highly depends on the *repeatability* of the edges. Given two images depicting the same scene, it is the

ratio of potential edge pixels associations over the number of edge pixels. This is computed by first detecting edges on the first image and then projecting them onto the second using the camera displacement and the scene depth. With n the number of pixels that falls into an edge in the second image and N the total number of edge pixels, repeatability is measured by $\frac{n}{\mathrm{N}}$.

Setup. Here, only N = 9000 edge pixels are randomly sampled from the first image for a fair comparison. This is motivated by the fact that the edge density has a high variance over the methods. This lead to values of N with a different order of magnitude.

The repeatability is measured on the simulated data vKITTI [58]. This prevents the noise of the camera motion or the depth estimation to influence the metric.

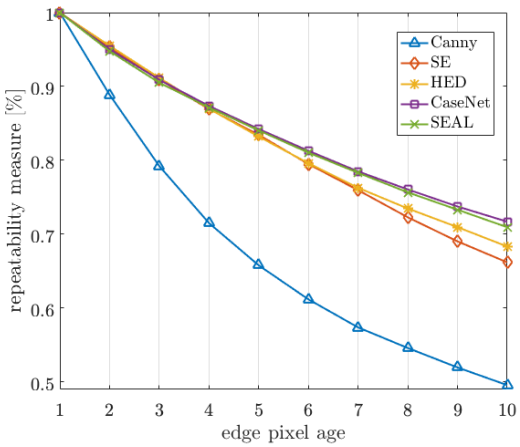


Figure 6.5: Repeatability analysis on vKITTI. We compare conventional edge detector (Canny [34]), learned edges (SE [47], HED [198]), and semantic edges (CaseNet [202], SEAL [203]).

Results. Fig. 6.5 shows the edge repeatability with respect to the number of frames between the two images. As expected, the repeatability decreases when the changes between the images increase. The results show that learned edges significantly outperform the conventional Canny detector, which justifies the recent effort on edge learning. Also, the plot suggests that semantic edges are slightly more robust than standard edge: their repeatability curves decrease slower than the latter's one.

6.3.4 Influence of the Edge Derivation

This section evaluates the influence of the derivation of learned edges for ANNF, ONNF and SNNF. Experiments show that end-to-end learning approaches provide semantic

edges better suited for visual tracking. Figure 6.6 visualizes the various edges.

Figure 6.6: Illustrations of the various edge detection methods.

Rationale. There are two main approaches to generate semantic edges. The first one relies on end-to-end learning approaches such as CaseNet and SEAL. Their input is a color image and the output is a set of probability maps, one for each class. Each map represents the probability of a pixel to belong to an edge in this class. The main differences are the training optimization and the emphasis put on the edge thinning.

The second approach is to fuse standard semantic maps with learned edges such as SE or HED. They take a color image and output an edge probability map of the same dimension. Each pixel value is the probability of that pixel to belong to an edge. This edge map is fused with a semantic probability map where each pixel value is the probability of this pixel to belong to a class. Such probabilities are computed from the softmax on the segmentation network logits i.e. the output of the last layer of the network. Finally, a map representing the probability of a pixel to belong to an edge with a given label is computed from the multiplication of the previous maps. This assumes that the edge variable and the label variable are independent, which is a reasonable assumption.

Setup. The semantic segmentation is computed with the Xception-65 [38] variant of the SoA DeepLabV3 [37]. It is pretrained on the Cityscapes dataset and an open implementation finetuned it on the KITTI dataset [13].

The pure edge-based methods ANNF and ONNF only integrate the geometric information of the learned edges, even when they are provided with semantics. Here, the photometric gradient constraint is integrated to the edge-based optimization.

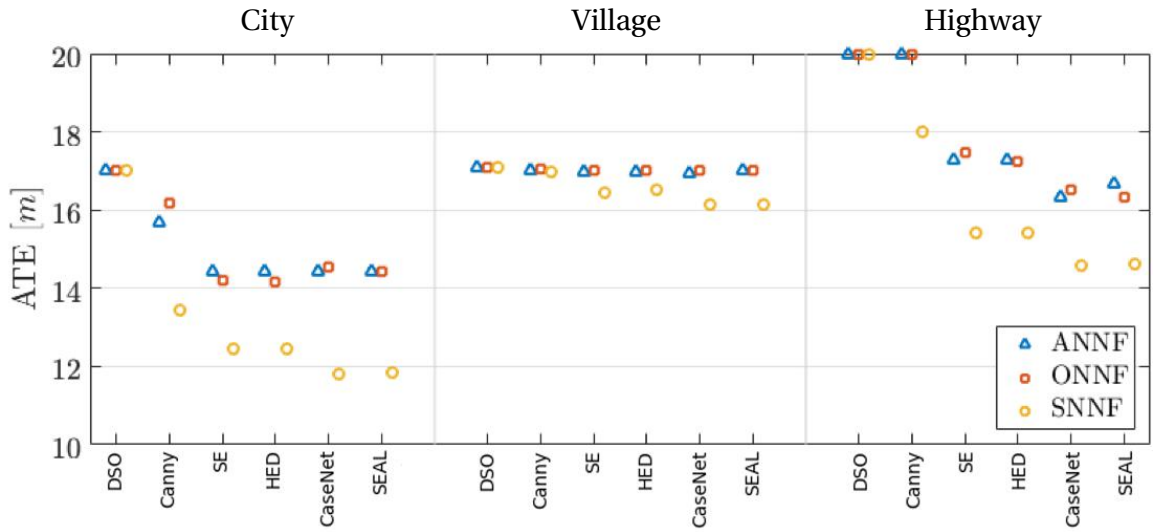


Figure 6.7: KITTI trajectory errors averaged over each environment (city, village, highway). The SoA DSO is compared to SNNF variants based on learned semantic edges, or the fusion and semantics and edges learned individually. The first approach provides better tracking results.

Results. Results suggest that semantic edges learned in an end-to-end manner are better suited for VO, whether the semantics are leverages or not. For a fixed edge detector, the semantic constraints lead to a lower ATE. This reinforces the relevance of semantics integration for visual localization.

Fig. 6.7 compares the influence of the edge generation on one hand the influence of the semantic integration on the other. The results are divided over the three KITTI environments: city, village, and highway. All edge-based approaches reach lower or equivalent error to DSO and the boost is significant for city and highway scenes. An intuitive explanation is that the more repeatable edges the image hold, the more constraints the optimization can leverage.

Even when ANNF and ONNF leverage the same geometric edges as SNNF, the latter achieves similar or lower ATE. This shows that the integration of semantic does benefit localization. Information is pivotal for robust tracking. The performance gap between the geometry-based ANNF and ONNF, and their semantic counterpart SNNF is significant when the semantics are accurate. This is the case for the city scenes for example. But when the segmentation gets noisy, the two approaches become equivalent (*e.g*, village). This result motivates the effort toward better segmentation models.

6.3.5 Runtime and Qualitative Results

When it comes to runtime performance, ORBSLAM2 and DSO stay the most competitive approaches. One main drawback of SNNF is the need for a CNN to generate the semantic edges. This makes the time for tracking and mapping 1.34× longer than DSO on average. This comparison is computed on the original KITTI and vKITTI image size ((1224, 376)px) for which the semantic edge generation takes 0.7 seconds per image on a NVIDIA 1080Ti. A relevant line of work that addresses this issue is model distillation [76]. Given a large CNN model trained on a visual task, a smaller model is trained to generate the same outputs as the larger one. This approach has already proven relevant for standard tasks such as classification, detection and segmentation [75, 76, 154], image retrieval [156] and local feature detection and description [155].

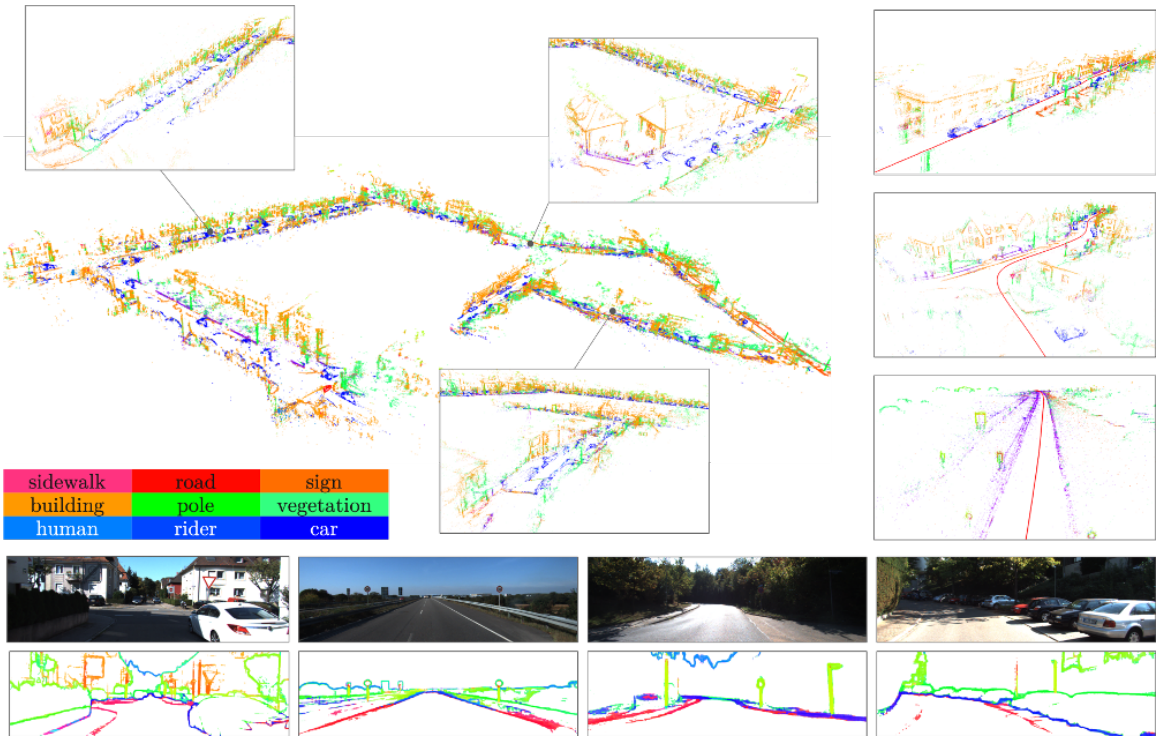


Figure 6.8: Reconstructed semantic edge maps for KITTI. Left: semantic edge maps recovered from city, village, and highway sequences. Right: semantic edge images generated using CaseNet [202].

6.4 Summary

This chapter studied how the integration of semantics into edge-based direct visual odometry addresses the tracking robustness challenge. It reinforces the edge pixel regis-

tration during the tracking step by constraining associated pixels to belong to the same semantic class. It takes into consideration that a pixel can have multiple labels and use these additional constraints to make the data association more robust. When compared to existing edge-based methods on an autonomous driving use case, it leads to lower trajectory error. This supports the idea that integrating semantics into existing localization applications can improve their performances and robustness.

This is not the first application where semantics augment a visual task: Chap. 3 and Chap. 4 already showed that fusing semantics with coarse of local geometry information defines global image descriptors that achieve SoA in multi-season scene retrieval. One issue with semantics is that they require large models trained with supervision on a high amount of data. This limits the integration of semantic in tasks with specific data and this is why the last part of this thesis discusses methods to reduce the training load of semantic segmentation.

Part IV

Semantics Training with Alternative Supervision

The previous parts of this thesis provide examples of how semantics can be integrated into visual localization applications to make them robust to image variations in appearance.

The state-of-the-art for semantic segmentation relies on large CNN models heavily trained on high amounts of specific data. Any time the pixel distribution of the data changes, CNNs usually fail to generalize well. For example, the segmentation network trained on the CMU-Seasons dataset outputs noisy results on the Symphony dataset, which limits the performance of the semantic-based localization. One solution is to finetune the CNN to the target application. But collecting supervision data is costly, time-consuming and sometimes even not feasible. This motivates the next part of this thesis to discuss methods that reduce the need for supervision data when finetuning segmentation CNN.

Chapter 7

Synthetic Data Generation for CNN Domain Adaptation[1]

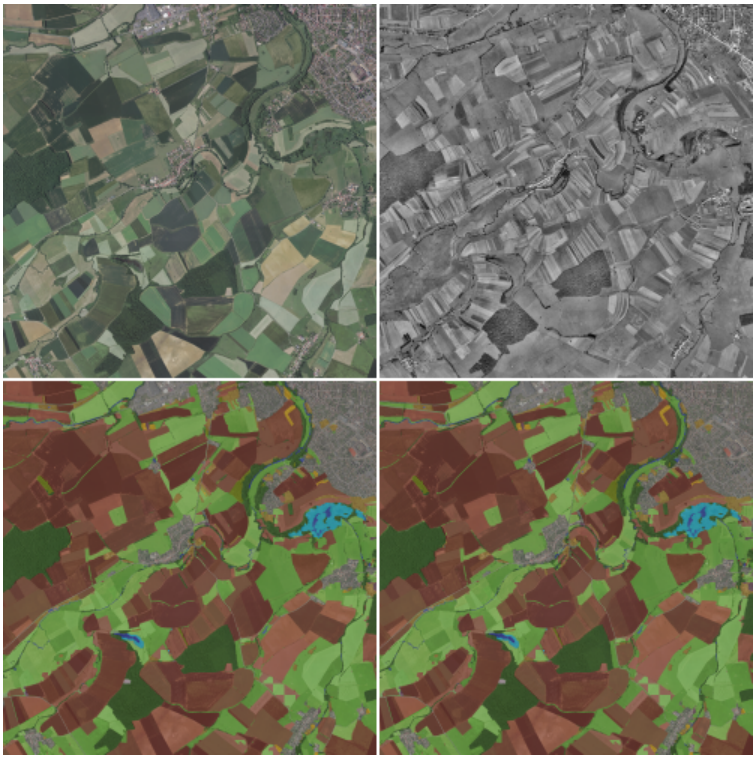


Figure 7.1: Two samples of the segmentation dataset. Left: the source data collected in 2015 with a digital camera. Right: the target data collected in 1955 with an analog camera and later digitized. Bottom: qualitative segmentation results.

The contributions presented in this chapter are the results of a collaboration with my Ph.D. colleague Antoine Richard and the master student Gabriel Hurtado. In this joint work, Antoine contributed to the data collection and processing, Gabriel contributed to

[1]This chapter describes contributions published in ICPA 2018 [148].

the synthetic data training and I contributed to the segmentation training. This work resulted in a paper that we jointly wrote and published in ICPA18 [148]. This chapter borrows some of the plots from the paper but the writing is my own.

Given a CNN trained with supervision on a source dataset, domain adaptation updates the network so that it generalizes on a target data. Figure 7.1 illustrates a segmentation example on two aerial images taken 50 years apart where each pixel must be classified according to the land type (*e.g*, fields, road, city). The goal is for the network trained on the recent color images (left) to generalize to the 1950 analog images (right).

One reason for poor generalization is when the source and target datasets exhibit different pixel distributions. The standard solution is to collect annotated images from the target data and finetune the network *i.e.* train it starting from the previous model optimized on the source images. However, this requires additional annotations that are costly and time-consuming. This chapter proposes to transform the source dataset so that its pixel distribution gets closer to the target one without changing its content. This way, the source annotation can be reused to finetune the CNN on this transformed data. By training the network on images with a closer distribution to the target's one, this method improves the CNN's generalization.

Figure 7.2: Neural Style Transfer Example [60]. Left: natural image. Right: the natural image with the painting style of the bottom image.

This approach is motivated by the neural style-transfer (Fig. 7.2) that modifies the image appearance but preserves the image's composition. The first approach from Gatys *et al.* [60] leverages the feature space generated by a trained CNN. It observes that two images with similar style exhibit the same feature statistics, and more specifically similar Gram matrices over the feature vectors. When the two images also have similar content, their feature maps are close in the Euclidean space. This allows neural-style-transfer to

find the minimal image update for the natural image to get similar feature statistics as the painting. The resulting image has the same content as before with the appearance of the painting.

The same approach is used to transform a source dataset to match the target data's statistics while preserving its content. Then, the segmentation annotations already available for the source dataset can be reused on the transformed images. The CNN is fine-tuned on the transformed data with standard supervision.

The rest of this section is organized as follows. Sec. 7.1 recalls the style transfer optimization. Sec. 7.2 introduces the segmentation dataset and the experimental results. The data is collected by an organization dedicated to long-term monitoring of national land occupation. Aerial images have been sampled every few years since 1950. The images present a wide range of appearance from recent color images taken with multi-spectral cameras to analog images that were later digitized. The experiments show that the synthetic data is not enough to achieve satisfying segmentation results. However, it proves useful to warm up the CNN. The network still needs to be trained on real annotated target data but need fewer examples and reaches higher segmentation performances.

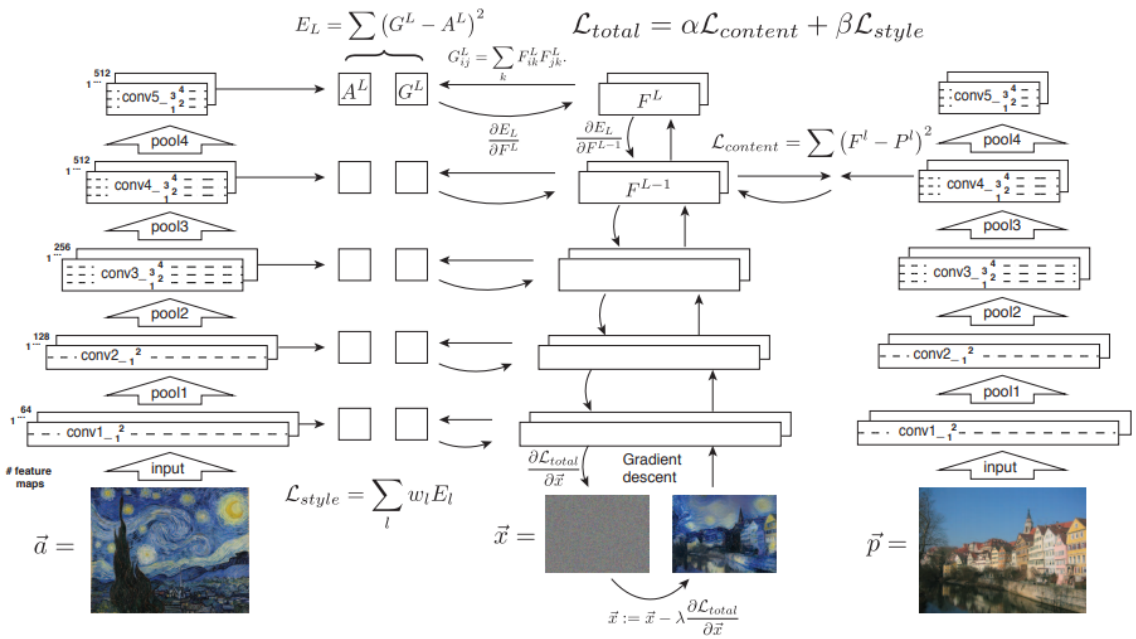


Figure 7.3: Illustration of the Neural Style Optimization in [60]. The same trained CNN is used to compute the loss from the feature statistics. The gradient of the loss with respect to the image is computed with backpropagation and used to update the image.

7.1 Synthetic Data Generation

This section describes the neural style derivation from Gatys *et al.* [60] and the speeded-up approach from Johnson *et al.* [86], which is the one used in the experiments.

The first neural approach [60], illustrated Fig. 7.3, relies on a trained neural network. The figure shows three networks but it is the same one replicated three times for the sake of clarity. Note that the network weights are never updated, only the image is. The style image (left) and the content image (right) are fed to the network and their feature maps are stored for later use. The goal is to generate an image so that its features share statistics with the style's ones but stay close to the content image's features.

The optimization starts from a random noise image fed to the network. This image is fed to the same network (center) and two losses are computed. The first is the content loss computed as the square residue between the features of the input and content images. The style loss is the squared difference between the Gram matrices of the style image and the input one. The Gram matrix represents the dependence of the feature vectors. One way to see a feature map F^l of size $h_l \times w_l \times c_l$ is a a set of $h_l w_l$ feature vectors of dimension c_l. The $(i, j) \in \mathbb{N}^2$ element of the Gram matrix is the dot product of the i^{th} and j^{th} feature vector. These two losses may be computed on different feature maps. For example, in Fig. 7.3, the style loss is computed on features from the 5^{th} convolutional block whereas the content one is computed from the 4^{th}. Each of the loss is backpropagated back to the image space to compute the gradient of the theses losses with respect to the input image. These two gradients are summed and integrated into an Stochastic Gradient Descent (SGD) optimization to update the image. These steps are repeated until convergence.

Johnson *et al.* [86] propose an alternative derivation to speed up the generation. They feed a network with the natural image and train it to output the transformed image. The training is also computationally demanding and a network must be trained for each style, but the generation at run-time is much faster. So this approach is preferred to the previous one.

7.2 Experiments

Dataset. The data to segment are overhead images collected in 1955 and 2015 over the Grand Est region of France. The French national Institute of Geographical and forestry Information (IGN) surveys the region regularly and provides rectified images ready to process for segmentation. The 2015 images are sampled using a multi-spectral camera

and only some channels are kept: red, blue, green and near-infrared. They exhibit a resolution of 50 cm per pixel so they can even represent narrow structures such as road and streams. The 1955 images are analog black and white photographs, later digitized and rectified afterward. These images still hold the same resolution but display some altering effects such as blur, grain, and saturation and poor contrast.

Human Annotation. Relevant land classes are defined by the EUNIS habitat classification [114] which is a pan-European system describing habitats across Europe. It gathers hundreds of labels and only 14 are kept for the sake of the monitoring application (Tab. 7.1). Most of them focus on wetlands such as hems and riparian groves. A human fuses various maps to annotate each pixels such as water network maps [3], forest maps [1], culture maps [2], and Google Street View. On average, it takes 8 hours. Given the monotony of the task, it is split over several days and in practice, it took the hired engineer a week to get the annotation for a 10 000 × 10 000 pixels image.

0	Encoder	7	Coniferous woodland
1	Surface standing waters	8	Tree farms
2	Constructed areas	9	Fruit orchards
3	Extractive Industrial Sites	10	Riparian vegetation
4	Grasslands	11	Heathlands, scrub and tundra
5	Arable lands	12	Chopping areas
6	Broadleaved woodland	13	Vineyards

Table 7.1: Semantic classes.

Data Processing. The data collected exhibits a strong class imbalance. For instance, there are 100 times more pixels of grasslands than woodlands pixels. Such a data pattern usually hinders the segmentation performance. A standard solution is data augmentation: it samples the images with under-represented labels and applies transformations that preserve the semantics (*e.g* rotations,crops). The resulting images are added to the training dataset. However, there is an additional constraint for the dataset used in this chapter. One image can hold both over and underrepresented classes. So the addition of such images would increase the class imbalance. To address this issue, an image is augmented only if it reduces the inter-class variance. Fig. 7.4 shows the class distribution before and after augmentation. The augmentation does not result in a uniform class distribution but it still improves over the previous imbalance.

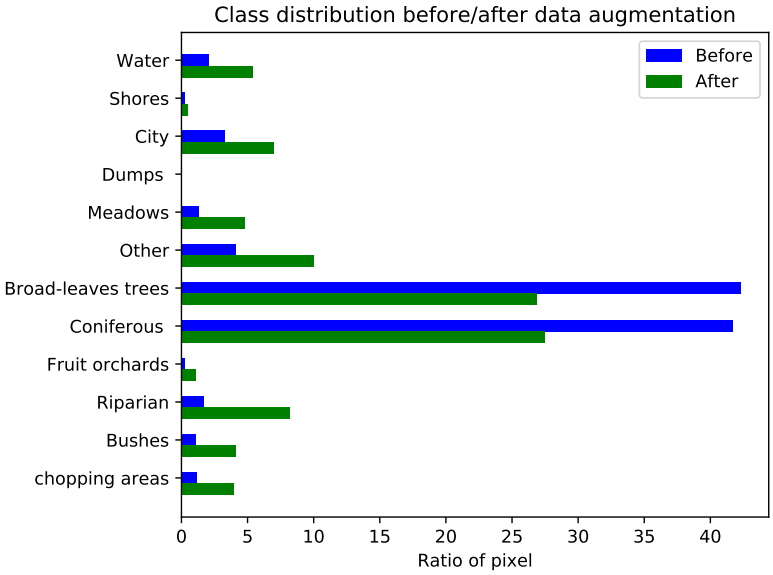


Figure 7.4: Pixel distribution of classes before and after the data augmentation. Although the augmentation does not compensate for the whole imbalance, it brings the classes closer to a uniform distribution than before.

Metrics. The segmentation model is evaluated with two standard metrics: the *accuracy* and the mean Intersection Over Union (*mIOU*).

The *accuracy* measures the ratio of pixels correctly classified. More formally, for a class c, TP is the number of pixels with correct classification, and FP is the number of pixels wrongly classified. The accuracy of class c is $acc = \frac{\text{TP}}{\text{TP+FP}}$.

The *mIOU* expresses how well the model locates the various semantic instances and how good it segments their boundaries. For a class c, let B_{gt} (resp. B) be the set of ground-truth (resp. learned) semantic boundaries. The *mIOU* is defined as $m\text{IOU} = \frac{\text{B}_{gt} \cap \text{B}}{\text{B}_{gt} \cup \text{B}}$.

7.2.1 Baseline: real supervision

Setup. Among the various segmentation models with SoA performance back in 2017 [18, 36, 112, 134], DeepLab [36] reached the highest accuracy and *mIOU* on the land dataset. The experiments use the Resnet-101 [74] variant of DeepLab pretrained on the PASCAL VOC [55] dataset. DeepLab fuses the segmentation of three independent Resnet-101 on three scales of the same image. Experiments showed that this multi-scale approach did not boost the performance for this specific dataset. So only the branch that processes the original scale is kept. This reduces the memory requirements from

8G to 3G of RAM. DeepLab is finetuned on the overhead dataset and the output is post-processed with an efficient implementation of Conditional Random Fields (CRF) [96]. This step denoises the segmentation, especially around boundaries.

The code is run using the Caffe [85] library on Nvidia GT1080 GPU set with Cuda8 and CudNN5. The segmentation is optimized with SGD [30] with an initial learning rate of $\alpha = 2.5 \times 10^{-4}$ and a polynomial decay. This reduces the learning every iteration using the formula $\alpha \rightarrow \alpha \times (1 - \frac{iteration}{30000})^{power}$ with $power = 0.9$. The SGD momentum is set to 0.9 and the weight decay to $5 \times 10{-}4$. The network is trained on 300×300 pixels image crops sampled from the original images.

Training Parameters. Table 7.2 summarises the parameters for the two separate trainings on first the 2015 data, then on the 1955 one.

	2015	1955
Train set size	15360	7328
Test set size	3232	790
Image size	300	300
Batch size	8	8
Train epochs	7	15
Training time	24h	24h

Table 7.2: Training parameters for the baseline segmentation.

Results. The DeepLab segmentation model trained on the 2015 dataset reaches 73% of accuracy in average and 75% of *mIOU*. It slightly underperforms on the 1955 data with only 65% of accuracy and 55% *mIOU*. The main explanation is that the first model is trained on twice as much images. Figure 7.5 details the *mIOU* for each class. Note that the classes with the lower scores are usually underrepresented, like the riparian vegetation.

Another cause for the network's confusion is the visual similarities in appearance between classes. For example, the riparian vegetation is visually akin to grasslands. The confusion is reinforced by the fact that there are many more examples of grasslands than riparian vegetation. When there are enough examples of two visually similar classes, the network manages to differentiate them. For the 1955 model, another source of error is the higher unbalance of the dataset than for the 2015 images. For example, only 35 images hold tree farms pixels after the data augmentation. This is another motivation

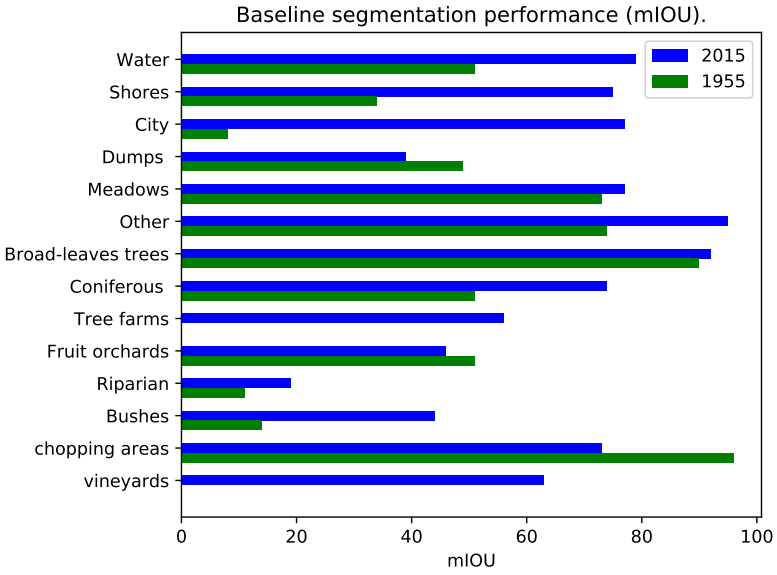


Figure 7.5: Segmentation *mIOU* performance for each class on the 2015 and the 1955 data.

to rely on the 2015 supervision as it is better balanced. This is the object of the next experiments.

7.2.2 Synthetic supervision

This section evaluates the segmentation trained on synthetic data. Experiments show such training achieves only a third of the performance reached with real data finetuning. So it needs to be additionally trained on real annotated images from the target data. Still, the network needs fewer images when it is warmed up with the synthetic data.

Baselines. The segmentation network is initialized with training weights on the source data *i.e.* the 2015 data. Then, it is finetuned using one of the three following 1955 datasets: the real 1955 images with human annotations, the synthetic 1955 images generated from all the 2015 data with the style transfer, and the black-and-white version of the 2015 images. The last set aims at showing that the style transformation is more than a change of the color domain. All the networks are evaluated with the segmentation performances on the real 1955 images.

Setup. The synthetic generation uses the Tensorflow [5] code released by Logan Engstrom [54]. The image transformation network follows the VGG network [171] and has

been previously trained for classification on ImageNet [98]. It is finetuned to generate the MS-COCO [110] images with the appearance of 1955 land images.

The segmentation optimization follows the same setup as for the training on the source data.

Training Parameters. The transformation network is trained with the Adam optimizer [93] with a learning rate of 10^{-3} for 2 epochs over the 80K COCO images. Each iteration has a batch of 4 images. The content loss is computed on the `relu_4_2` feature map and is weighted with $\lambda_w = 7.5$. The style loss uses the `relu1_1, relu2_1, relu3_1, relu4_1, relu5_1` maps with a weight of $\lambda_{style} = 10^2$. The generated images are regularized with total variation regularization with a weight $\lambda_{reg} = 2 \times 10^2$.

Data	Accuracy (%)	*mIOU* (%)
Black-and-White 2015	10	4
Stylized 2015	24	12
Real 1955	**65**	**55**

Table 7.3: Segmentation results on the real 1955 dataset. The networks, previously trained on the 2015, are finetuned on either the black-and-white 2015 data, the stylized 2015 data, or the real 1955 data. Only the finetuning on real 1955 data converges.

Results. Finetuning on the synthetic achieves only a third of the performance of real data finetuning. Tab. 7.3 shows that the synthetic data achieves no more than 24% accuracy whereas finetuning on real data gives the top score (65%). The stylized data leads to better results than the black-and-white one, which shows that style transfer runs more than a simple color change. However, it is not enough to bridge the gap between pixel distributions of the source and the target data.

Data / Network	DeepLab	SegNet
Back-and-White 2015 + Real 1955	67.28	63.60
Stylized 2015 + Real 1955	**70.37**	**64.28**
Real 1955	65.0	NA

Table 7.4: Segmentation boost measured with mean accuracy (%). The segmentation networks previously trained on 2015 images are finetuned on real 1955 images. The first two lines show the performance when the CNN is first finetuned on synthetic data then on the real one. This additional training boosts the segmentation results, especially with the stylized data.

Still, synthetic data seems to be relevant to warm-up the CNN on the target data (Tab. 7.4). This agrees with previous observations in the litterature and motivates the generation of high-definition synthetic datasets [149, 150]. The segmentation networks previously trained on 2015 images are now finetuned on synthetic data before the real 1955 images. This boosts the segmentation performance from 2% to 5%, which is a significant boost for segmentation. Even the simple black-and-white transformation allows for a boost of 2% accuracy.

The same test is run on SegNet [18] too, another SoA segmentation network (Tab. 7.4 - right). In this case, finetuning on the real 1955 data only does not allow SegNet to converge. One possible explanation is that SegNet runs a weighted pixel-wise classification where the weights are inversely proportional to the frequency of this class in the dataset. When the unbalance is too extreme, as in the 1955 dataset, this induces weights with high amplitude that hinders the convergence. In this case, warming up the network provides a better starting point in the optimization space. Note that for SegNet, the simple black-and-white images provide a warm-up almost as useful as the styled images.

7.3 Summary

This chapter described how to leverage previous contributions in neural style transfer to generate synthetic data for segmentation adaptation. It assumes that a source dataset is provided with annotation. These source images are transformed so that their pixel distribution gets closer to the target distribution while preserving their content. This is achieved using the style transfer optimization. The CNN is then finetuned on the transformed data with the source annotations. Experiments show that this optimization achieves only a third of the performance reached with real data. However, this synthetic data proves to be useful to warm up the CNN. Then finetuning on the few real target images achieves better results than without the warm-up.

Chapter 8

Features Transfer for Segmentation Adaptation [1]

The previous chapter proposed to transform the distribution of an annotated source dataset to fit the 'style' or distribution of the target data. Then a CNN trained on the transformed data should better generalize on the target one. Experiments show that the transformed data is useful to warm up the CNN but the training still needs real annotated images from the target dataset to provide satisfying performances. Still, this provided some insight on the information inside the feature maps: they not only hold representations of the image content but also statistical information about the image appearance, what was called the 'image style'. And it showed that it possible to transfer the 'style' of one image to another using this feature information. In other words, the pixel distribution of a dataset can be changed to get closer to another dataset. One issue is that the computed optimization to transform the data is computationally expensive and time-consuming. So this chapter investigates a simpler way to leverage the information embedded in the CNN feature maps to transfer image distributions from one dataset to another.

Previously, the style of the target dataset was captured by the features of the CNN trained on the source dataset. It was brought back to the image space to transform the source data. This transformed data was sent back to the feature space of a distinct CNN to finetune it on the target data. Instead, this chapter proposes to short-cut this process and only manipulate feature information. One drawback of the proposed method is that it requires semi-annotations in the form of approximately aligned images.

The target CNN is initialized with the weights of the source CNN trained on the annotated data. It is then trained to generate the same feature maps on the target data as

[1] This chapter describes contributions published in ICONIP 2019 [27].

the source CNN did on a source image. The constraint is that the source image holds approximately the same content as the target one. The advantage is that there is no need to generate new data and finetuning becomes a simple regression problem.

The rest of this section is organized as follows. Sec. 8.1 derives the training approach. Sec. 8.2 presents both quantitative and qualitative results: the proposed method reaches performances similar to classic transfer learning on the PASCAL VOC dataset with synthetic transformations. And feature visualization suggests that the target CNN now projects both source and target image into the same point in its feature space. This means that it has become sensitive only to the image content and not to the image pixel distribution. This is what is expected from aligned images but with different pixel distributions after finetuning.

8.1 Domain adaptation from feature map regression

8.1.1 Feature map regression

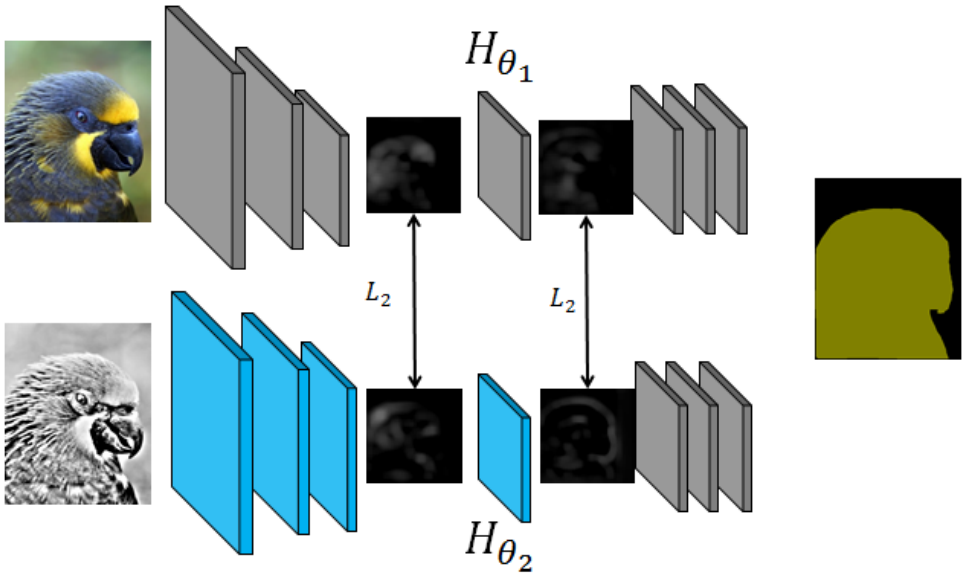


Figure 8.1: Top: The trained and frozen (gray) network provides ground truth deep representations. Down: The trainable layers (blue) must learn the deep representations.

In Figure 8.1, the top network is the source network trained in a supervised manner on an annotated source dataset D_s. After training, it is frozen (grey) and its feature maps hold high-level semantic representations of the image content.

The bottom network is initialized with the source weights and is trained on the target dataset D_t. The training requires a pair of approximately aligned images with different

pixel distributions $(X_s, X_t) \in D_s \times D_t$. The target network H_t is adapted to generate the same features maps on X_t as the upper network H_s on X_s. This is achieved with a simple regression between the feature maps of the layers to adapt (blue). Note that this process does not require segmentation annotation on the target data D_t. The target network is trained by backpropagating the Euclidean distance between analog feature maps of the two networks. The difference between the l^{th} feature maps is backpropagated through the previous layers only. This leaves the option to adapt only a subset of feature maps rather than the whole target network.

8.1.2 Visualising the Feature Adaptation

This section adapts existing visualization techniques [60, 117] to try to observe the feature maps evolution. A source image X_s is fed to the target network H_t to generates a set of l feature maps $\{F_t^l(X_s)\}_l$. The goal is to invert this features back to image space *i.e.* generate an image X that leads to the features $\{F_t(X_s)\}_l$. The goal is to visualize how the network sees the image.

X is initialized with white noise and is optimized so that $F_t^l(X) = F_t^l(X_s)$ for all feature maps. This is achieved by backpropagating all the feature map residues back to image space. The result is the accumulation of image gradients from all the feature map errors. The input X is updated with this image gradient using SGD [117].

Previous work on neural style transfer by Gatys *et al.* [60] observed that the previous optimization only constrains X in content and not in style or distribution. This constraint is expressed with the Gram matrix: $G_t^l(X_s)$ designates the Gram matrix of the l^{th} feature map of the target network when fed with a source image X_s. In addition to the previous content condition, X should also satisfy $G_t^l(X_s) = G_t^l(X)$ for all feature maps. As for the feature map difference, the Gram residues are backpropagated into the image space. Then X is updated with these cumulated gradients using SGD. See [60] for more details on the Gram matrix derivation.

8.2 Experiments: semantic segmentation adaptation on PASCAL VOC

Metrics The method is evaluated on the segmentation performances of the target network on the target data. The standard segmentation metrics defined in the previous chapter are used again: the accuracy and the *mIOU*.

Dataset The experiments are run on three synthetic transformations of the augmented version [70] of the PASCAL VOC12 dataset [55]. It holds 10 582 training images and 1449 validation images with 21 semantic classes. The original dataset contains 1464 train images, 1449 validation images, and 1456 test images. The regression is trained on the 10 582 original images.

Three transformations T^1, T^2, T^3 with increasing perturbations are generated with GIMP [2] resulting in the three target datasets D_t^1, D_t^2, D_t^3 (Figure 8.2). The 'photocopy' filter T^1 emulates a change of color and saturation. This problem arises in long-term environmental monitoring where recent datasets are numerical RGB images and older datasets are collected with the numerization of analogic pictures [148]. The ripple distortion T^2 simulates image misalignment and edge noise. This is typical in natural environmental monitoring such as in the dataset from [66]. Finally, texture and edge noise are mixed with the 'cubism' filter T^3.

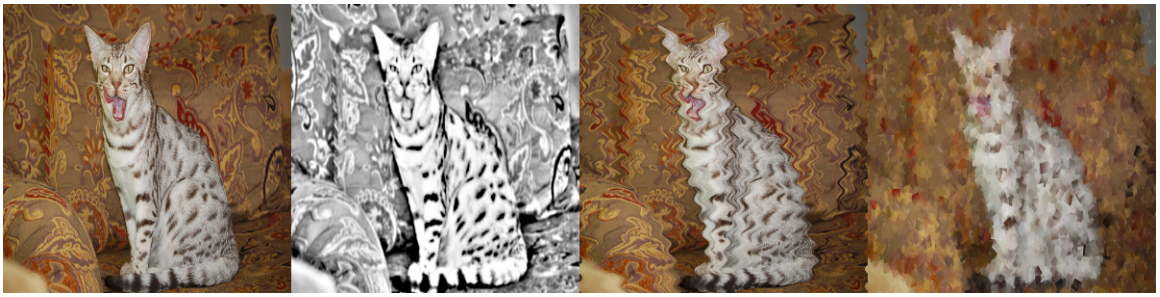


Figure 8.2: Synthetic transformations. Column 0: PASCAL. Left-Right: transformation. Photocopy (Distortion: 32.5%), Ripple (62.6%), Cubism (94.0%)

For each transformation, the image distortion between the source dataset and the target one is quantified with the performance degradation of the source network on the target data. In the experiments, the network uses the DeepLab V3 architecture [36]. After training on the source images, the accuracy and mIOU reach respectively 79.92% and 69.22%. So the image distortion is quantified by:

$$\frac{1}{2}\left(\frac{|79.92 - \text{accuracy}(H_s, D_t)|}{79.92} + \frac{|69.22 - \text{mIOU}(H_s, D_t)|}{69.22}\right) \tag{8.1}$$

T^1: Photocopy	T^2: Ripple	T^3: Cubism
32.48	62.59	94.03

Table 8.1: Quantification of the dataset distortion (%).

[2] https://www.gimp.org/

The dataset distortion values (Tab. 8.1) follow the visual intuition that the three transformations exhibit an increasing level of complexity.

Setup.

Supervised training of the source network. Both networks use the VGG-16 architecture [171] from DeepLabV3 [36]. The source network is trained on the PASCAL dataset during 5 hours on an NVIDIA 1080Ti with the original optimization parameters. The network is trained for 20 000 iterations with a batch size of 10, SGD with a momentum of 0.9, a weight decay of 0.5 and the polynomial learning rate policy initialized at 2.5×10^{-4} and $power = 0.9$.

This setup is used for training the source and target networks on their respective dataset with supervision.

Feature map regression on the target network. The choice of feature map to regress is investigated in the experiments.

The target network is initialized with the source weights. Then it is trained for 20 000 iterations with SGD with a momentum of 0.9. The learning rate is initialized at 10^-4 and decreased at each step following the polynomial policy with a power of 0.99.

8.2.1 Comparison with the Baselines

B	Training	Test
B_0	H_s on annotated D_s	H_s on D_t
B_1	H_t on annotated D_t	H_t on D_t
B_2	H_t initialized with H_s, fine-tuned on annotated D_t	H_t on D_t

Table 8.2: Baselines summary.

Baselines. The feature regression is evaluated with the segmentation performance of the target network on the target data. It is compared against three baselines (Table 8.2).

The baseline B_0 measures the performance of the source network H_s on the target dataset D_t, i.e. how well the source network generalizes to the target dataset.

The baseline B_1 trains the target network with full supervision on the target data D_t using the annotations from the source data D_s. This is an ideal training setting.

The last baseline B_2 measures the performance of the target network when it is initialized with the source network and then finetuned in the standard way. This evaluates the classic supervised fine-tuning and sets the performance the regression approach should reach.

Setup.

Ideal finetuning. The target network is initialized with the source weights. Then it is trained on the target data with the same optimization parameters as for the source training.

Feature map regression on the target network. Here, the regression adapts all the network layers up to the fifth block of VGG convolution. The loss is the Euclidean distance between the `pool5` outputs of the source and target networks.

Results The regression adaptation reaches similar or higher performance than classic supervised fine-tuning.

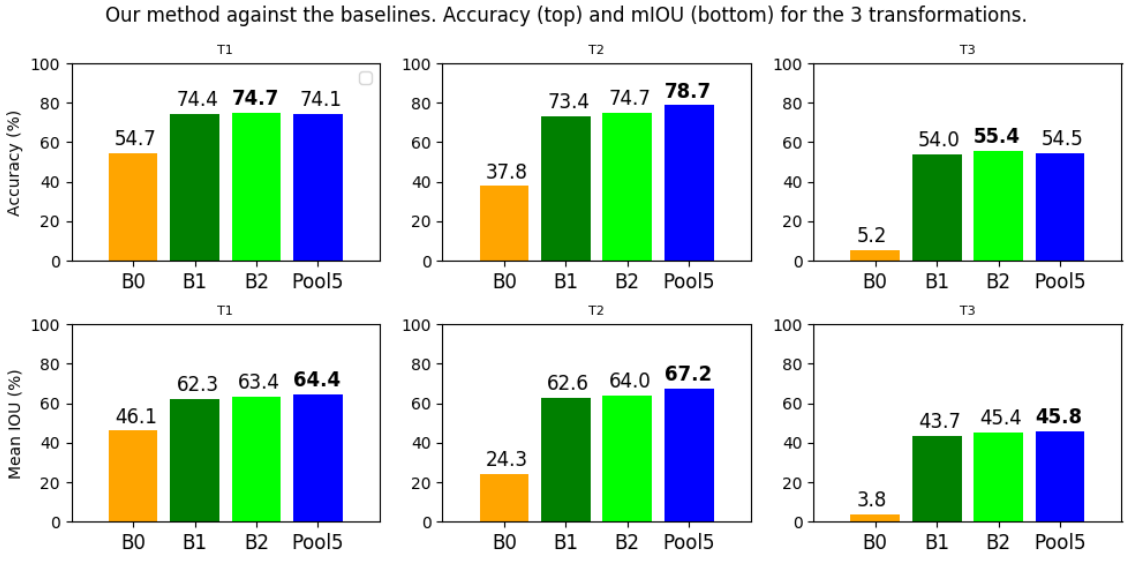


Figure 8.3: Transfer performance against the baselines.

Fig. 8.3 shows that the regression on `pool5` reaches similar performances to classic fine-tuning. The B_0 line recalls the performance of the source training on the target dataset. The regression adaptation improves the segmentation so this makes it a relevant fine-tuning method.

As expected, classic finetuning B_2 of the target network outperforms the cold training of this network on the target data. This reinforces the importance of network warm-up [102].

The regression adaptation achieves better *mIOU* than standard finetuning and similar accuracy. This shows that feature transfer is a relevant adaptation alternative for segmentation. Another advantage is that the supervision by the regression approach is simpler: sampling approximately aligned images is easier than pixel-wise labeling. Still, further experiments are needed to compare it with other works in domain adaptation.

8.2.2 Influence of the regressed feature map.

Setup The regression adaptation is run subsets of the target network layers to better understand the CNN's feature hierarchy.

An intuition gathered from the literature [48, 136, 169] suggests that early layers capture low-level representations such as colors and edges, whereas higher layers embed more complex features such as object contours and their label. This intuition suggests that adapting high layers is more relevant than lower ones as the image transformation gets important. This assumption is tested by adapting subsets of the network layers. A natural split is to adapt the VGG convolution blocks: for example, the convolutions up to `pool1` or up to `pool3` are adapted. The output of pool layers is preferred to the output of convolutions because they exhibit higher visual changes between the source and target data. When looking at the VGG feature maps, the features of successive convolutional layers look highly similar whereas there is always a break in the visual patterns after the pooling layers.

This experiment also assesses the correlations between the network's feature maps. Rather than backpropagating the loss between one feature map down the network, several Euclidean losses are computed at successive levels of the network. For the loss between the features after `pool1` is backpropagated down the first convolution block and the same goes for higher blocks. Once again, only the loss between post-pooling layers is computed. Two weight strategies are tested: the first one gives more weight to lower-level layers and the second does the opposite. There are five feature maps post-pooling and the weights follow the following weight distribution $[0.2, 0.4, 0.6, 0.8, 0.9]$.

Results Fig. 8.4 compares the segmentation performances with respect to the adapted layers. The best performances are reached with the individual regression on the highest post-pooling layer `pool5`. This suggests that high-level representations are the most relevant to transfer for semantic segmentation.

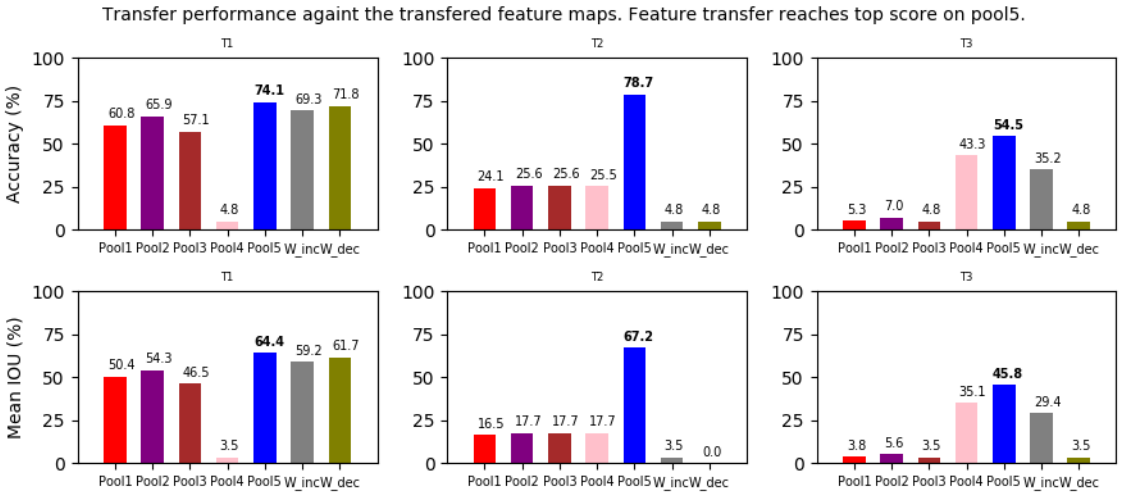


Figure 8.4: Transfer performance with respect to the transferred features maps. Transferring on `pool5` gives the best scores. Left-Right: Photocopy, Ripple, and Cubism transformations. Adapting all layers to adjust the feature post-`pool5` gives the best scores.

The experiments suggest that there is a correlation between the type of image transformation and the relevant layers to adapt. When the image undergoes color or saturation variation, like the photocopy transformation, adapting low-level layers gives reasonable results. This aligns with the common assumption that color processing is handled in the low-level layers of a CNN. When the image edges are modified, higher layers must be adapted. This also agrees with the hypothesis that contours are processed in higher layers than for color.

Another observation is that over-constraining the features may be counter-productive. For example, the transfer learning on multiple layers performs worse than the transfer on `pool5` only. In the cubism experiments, the results are better when the regression on higher layers is given more weights than the opposite. This is surpising as recent work has shown that multi-level optimization can speed-up the network training and even make it more robust as in [49, 77, 122, 202]. Further work is needed to better investigate the importance of various feature levels in the network generalization.

8.2.3 Visualisation of the features adaptation.

Setup A source image X_s is fed to the target network H_t to generates a set of l feature maps $\{F_t^l(X_s)\}_l$. The goal is to invert these features back to image space *i.e.* generate an image X that leads to the features $\{F_t(X_s)\}_l$ according to the target network. The goal is to visualize how the network sees the image.

The image X is constrained by the content and the style losses over the output of the first convolution of each VGG block. The loss computed from the feature map `conv_i_1` are backpropagated only to the lower convolutional blocks until the image space. The style loss is given a higher weight than the content one: $w_{style} = 10^-2$ and $w_{content} = 10^-4$. The losses on the feature maps from blocks higher than the first one are down-weighted by a factor 0.1. The optimization runs for 3500 iterations using SGD with momentum 0.9 and learning rate 10^-2.

Results Figure 8.5 shows that the images reconstructed from the target network all exhibit the style the network is adapted on, even when the input image has no specific style. This suggests that two images with the same content but different styles are projected to the same point in the target network representation space.

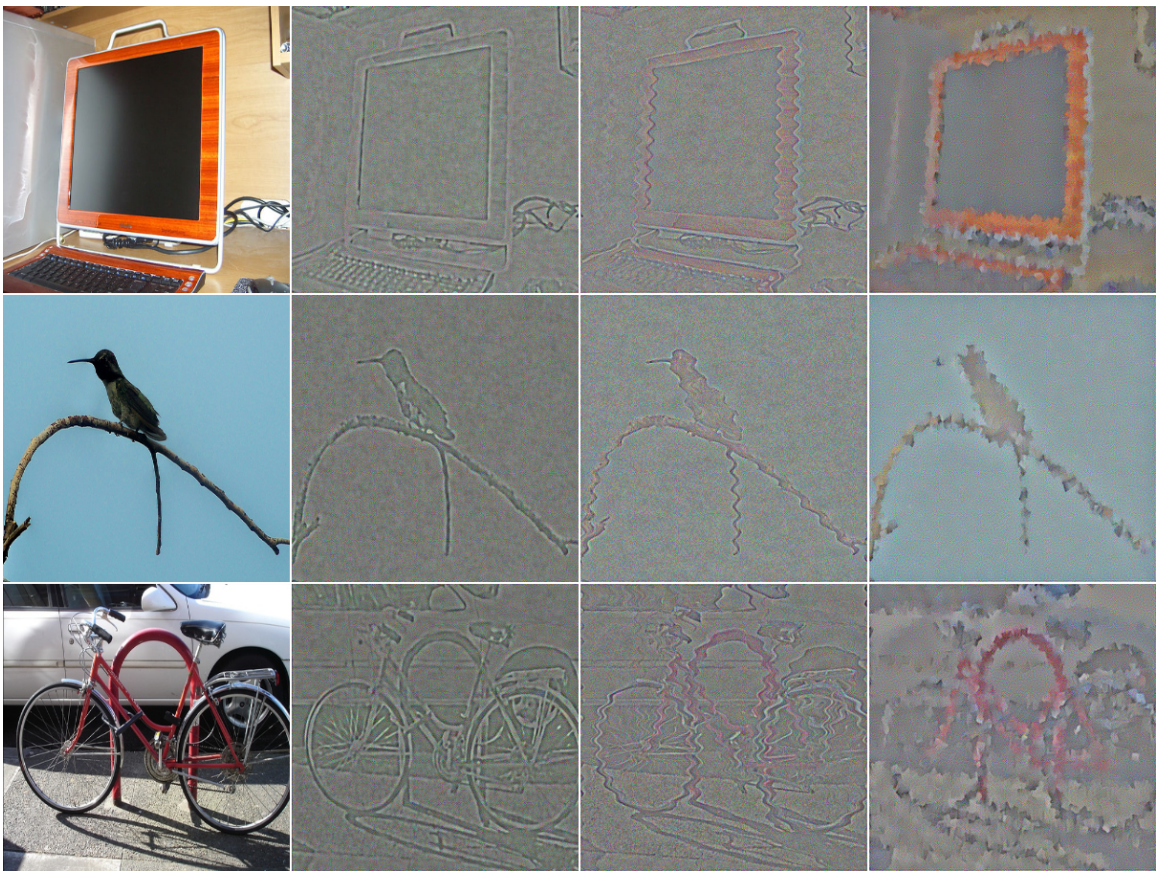


Figure 8.5: **Image reconstruction.** Left: source image fed to the target network. Right: The image reconstructed from the network feature maps *i.e* the image as seen by the network. The generated images have the same content as the source image but the style of the adapted network.

Figure 8.6 visualizes how target feature maps get visually closer to the source features after the regression.

8.3 Summary

This chapter introduced an optimization to adapt a segmentation network to a novel image distribution with operations run only in the feature space. Compared to the previous chapter, this has the advantage to reduce computational complexity. But this comes at the price of semi-supervision in the form of approximately aligned images. Given a set of aligned images with two different distributions, the target network is trained to generate a feature space that is invariant to the distribution changes. This method achieves the same performance as standard finetuning on three synthetic transformations of the PASCAL dataset. Experiments suggest that there is a correlation between the distribution variations and the optimal layers to adapt. Future work will investigate this correlation to improve adaptation performances.

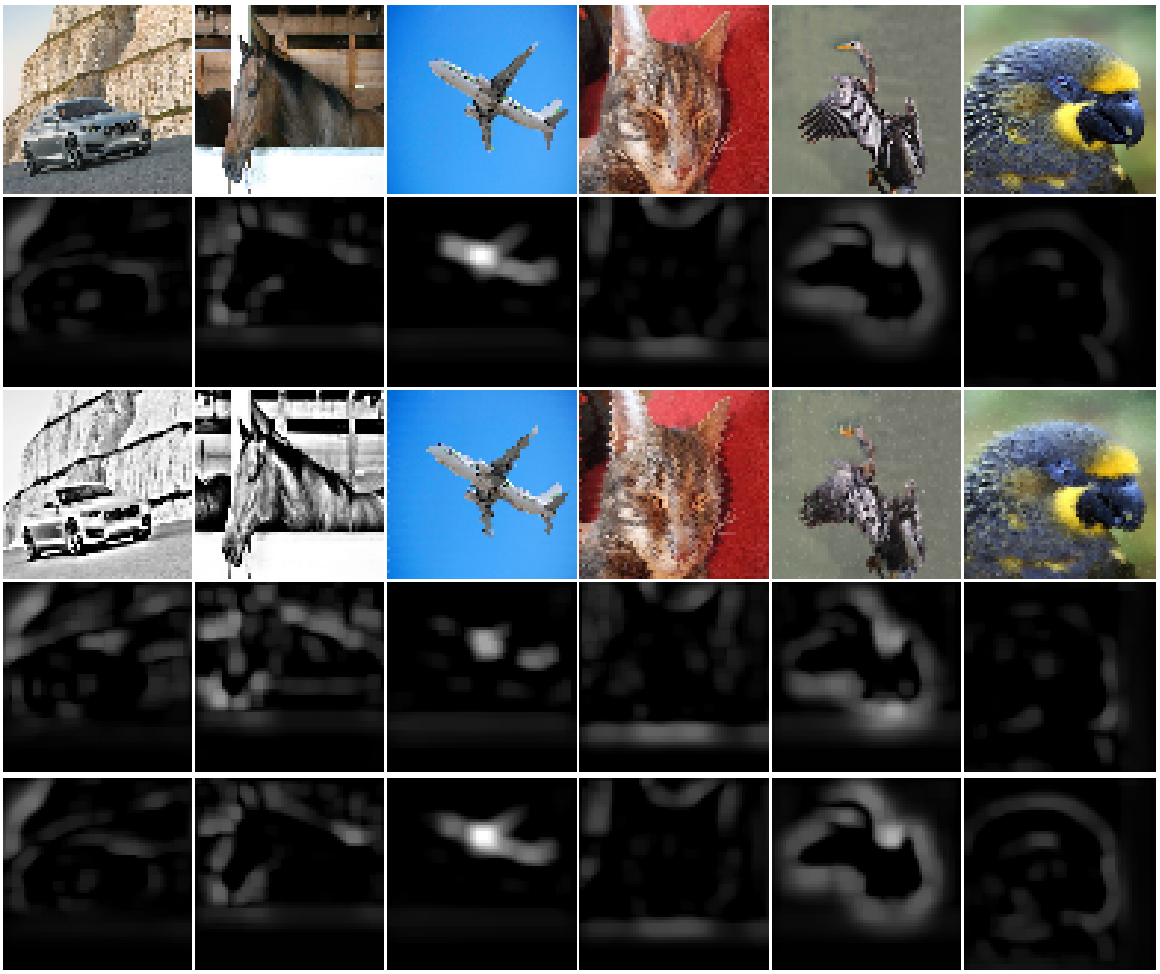


Figure 8.6: **Deep representation evolution.** Line 1: source image. Line 2: source feature. Line 3: target image. Line 4: feature of the source network on the target image. Line 5: feature of the adapted network on the target image. The last line shows the network features on the target images after the regression adaption. The features appear less noisy than before the adaptation (line above).

Chapter 9

Conclusion

This thesis has considered the robustness problem of visual features for long-term image-based monitoring in bucolic environments. This task involves various types of features for the scene recognition step, the localization of the monitoring system, the image alignment, and the pixel-to-pixel image comparison. One common requirement for these features is the invariance to image variations in appearance that are not related to the scene content or structure. In place recognition, for example, two images of the same scene should have features invariant to seasonal alterations. This way, a system can recognize a location previously visited by comparing its current image description with the past ones. This requirement also benefits semantic segmentation. For example, an autonomous car should be able to recognize a tree whether it is spring or autumn. This thesis relies on several approaches to define robust features. The first one integrates semantics in the feature derivation for place recognition. The second one leverages the feature space of neural networks to get data-specific features without training. Semantics are not only useful to define invariant features but also to robustify existing localization pipelines such as edge-based visual odometry. Given the importance of semantics, this thesis investigates approaches to reduce the need for supervision for segmentation and facilitate its usage in real applications. The next sections summarize this thesis and present the research problems that it identified.

9.1 Summary

Global Description. The first part of this thesis addressed the problem of image-based scene recognition in bucolic environments. Place recognition is the process by which a place that has been observed before can be identified when revisited. In practice, it is usually cast as an image retrieval task where a query image is matched to the most similar image available in a database. The search is computed on a representation of

the image content on a lower-dimensional space. The challenge is then to compute a compact and robust image encoding such that images of the same location are near to each other despite their change of appearance due to environmental changes. Most of the existing approaches are tailored for urban settings, in which pictures exhibit rich semantics or strong textures. Instead, this thesis tackles the problem for images depicting nature or structures with few semantics or textured elements. The type of the variations is different and that leads to a different approach to describing a bucolic scene that integrates semantics and geometry. The two descriptors, WASABI [23] [1] and SG-VLAD [25] [2] [3] achieve competitive performance on two multi-season bucolic environments: Extended-CMU-Seasons-park and Symphony-lake. They also generalize to urban scenes and reach performance similar to the current deep baselines NetVLAD and DELF.

The standard derivation of most of the global description follows three steps: the detection of characteriztic local elements, the description of these local elements, and their aggregation into a global description. In **Chap. 3**, WASABI selects semantic edges as relevant locations and describes them with their wavelet transform over a fixed-sized subsampling of the edge. These local descriptors are aggregated by simple concatenation. Then, the distance between the two images is the cumulative distance between associated edges. An edge from the first image is associated with the nearest edge with the same semantic label in the other image based on their descriptor distance. In theory, the Wavelet transform decomposes a signal uniquely over a basis of wave functions. This makes an ideal descriptor as it defines a bijection between an edge signal and its wavelet transform. In practice, this property does not hold because of the precision limits of the computer: image edges are discretized and only a subset of the wavelet coefficients are kept to reduce the memory requirements. This descriptor stays reliable enough to achieve state-of-the-art performance in place recognition. Still, further investigation on edge descriptions could boost the recognition results.

WASABI is evaluated on two multi-season dataset: the CMU-Seasons and the Symphony datasets. The CMU-Seasons dataset illustrates an autonomous driving scenario with 12 car traversals recorded over a year. It depicts images from Pittsburg's city and parks. The Symphony dataset illustrates a long-term monitoring scenarios with 141 autonomous boat traversals along the shore of the Symphony lake. It depicts vegetation at

[1] WASABI code: https://github.com/abenbihi/wasabi.git
[2] SG-VLAD illustrative video: https://youtu.be/JeYpcRPqDUM
[3] SG-VLAD code: github.com/W2desc/wasabi2.git

various states with poor texture and few semantics. WASABI is compared against handcrafted and learning methods and the latter are finetuned to the bucolic environments when possibles. In the bucolic settings, this novel descriptor achieves better recognition performance than previous approaches and the results are consistent over the seasons. This shows that this representation is invariant to such variations. Further experiments demonstrate that WASABI generalizes to urban scenes and even reaches similar performance as one of the current baselines, NetVLAD. Note that it does not need to be trained or tuned to do so contrary to the learning approaches. WASABI only relies on the availability of CNNs previously trained for semantic segmentation, which are widely made publicly available. However, research on unsupervised training of segmentation would broaden its range of applications.

While WASABI achieves better performance than previous methods, it presents three main limits, addressed in **Chap. 4**, that hinders its integration in real systems. First, it relies only on *coarse edge description*, so it ignores the edge's local variations that could further characterize the edge. Second, *the image representation is not scalable* with the number of edges. The image descriptor is the collection of the semantic edge descriptions. So, its size increases with the number of semantic edges. Similarly, *the image distance is not scalable either*. The distance between the two images is the cumulative distance between their matching edge descriptors. Although successful, a better distance computation would avoid the loop over all edges. This motivates the definition of second image descriptor, dubbed SG-VLAD, to tackle these limits.

SG-VLAD also follows the standard retrieval pipeline, that is local feature detection, description, and aggregation. Local features are the scale-space acceleration maxima of semantic edges. They are described with the edges' local variations derived with Shape Context Descriptor [22]. Now, local features have a finer resolution than the whole edge. They are aggregated by augmenting the existing VLAD aggregation with semantics constraints. The resulting descriptor represents the distribution of these local features over a dictionary of semantic visual words previously computed. The descriptor is made of the cumulative residues between each local feature and the nearest visual word, accumulated over the words. Beforehand, a collection of visual words is computed by clustering these local features over training images. This derivation is modified to integrate semantics: local features are first grouped by semantics before intra-class clustering. At the time of image description, local features are assigned to the nearest visual word with the additional constraint that they must belong to the same semantic class. Note that the image description has now a fixed size for all images that only depends on the number of words and the dimension of the local descriptor. The scale and orientation

discretization chosen for the Shape Descriptor lead to a local descriptor with size 48 and the number of visual words is chosen depending on the image content. More words are used for the semantic classes that occur often such as the vegetation in the CMU-Park or the Symphony lake, or the buildings in the CMU city. In total, experiments use no more than 40 visual words. The resulting descriptor is four times smaller than NetVLAD and DELF before their dimensionality reduction step. All these modifications allow SG-VLAD to address all the limits of WASABI and experiments show that it achieves similar performance on CMU-Seasons and significantly boosts the results on Symphony.

Local Description. In the second part of this thesis, another approach to local feature definition is addressed. **Chap. 5** introduces a novel local detector, dubbed ELF [24] [4] [5], based only on information embedded inside a CNN already trained on standard learning tasks, such as classification, with no further training. Previous works on deep local features already take advantage of representation properties to generate local descriptors [57, 113]. Instead, ELF extracts feature locations from the network to build a detector. This information is computed from the gradient of the feature map's norm with respect to the input image. The output is a saliency map with local maxima on relevant keypoint locations. Contrary to recent CNN-based methods, this requires neither supervised training nor finetuning. The detected keypoints are then described by the same CNN's features. A descriptor is generated by interpolating of one the network's feature map on the keypoint's location. This is the same derivation as in UCN [40] later used in SoA SuperPoint [46] and D2-Net [51]. ELF differs in that it relies on a CNN previously trained on a standard vision task whereas they they trained the CNN specifically for description.

ELF is tested on three classification networks and on two feature networks: SuperPoint and LF-Net [135]. It is evaluated on the repeatability of the detected keypoint and the matchability of the local features. When compared against the main hand-crafted and learning approaches, ELF achieves similar performance. Surprisingly, the ELF derivation applied to LF-Net even reached higher numerical results than LF-Net. However, LF-Net exhibits better robustness to image rotations. The ELF detector can be integrated with existing local descriptors, whether hand-crafted or learned. This usually improves the matching results.

While ELF compares to existing methods in terms of matching score, these prove to be better suited for structure-based localization. Local matches are fed to a SfM pipeline

[4] ELF code: https://github.com/abenbihi/elf
[5] ELF demonstration video: https://youtu.be/oxbG5162yDs

to reconstruct the 3D of the scene. Given a query images, its local features are matched with the ones from the 3D scene. These 2D-3D correspondences are then used to recover the camera pose of the query image. When evaluated on the feature matching challenge [6] and the day-night localization challenge [7] at CVPR 2019, ELF underperformed compared to other methods such as D2-Net or SuperPoint. It is not easy to pinpoint what hindered the results given than the localization pipeline integrates several components. Two elements are being investigated: the sparsity of the ELF detection and its localization precision. Another limit is that there is currently no automatic criteria to select the feature map from which the saliency is derived. A visual criterion has been established that favors the feature maps for which saliency exhibits sparse high-intensity signals. A numerical formulation of this criteria is under investigation.

Semantics for robust localization. The third part of this thesis returns to the use of semantics for robust visual localization. **Chap. 6** illustrates an example where semantics are integrated into monocular direct visual odometry. Direct VO allows a system mounted with a camera to recover its trajectory inside an estimated map of the world using images only. Given two successive images, it jointly optimizes the image's depth and their relative camera displacement until they can be perfectly warped over each other. The optimization is assessed with the pixel-to-pixel difference of the aligned images, which assumes that the brightness is constant. It is mostly true in indoor environments for which direct VO achieves impressive results, but it is not the case outdoor.

So rather than aligning pixels, edge-based VO proposes to align edges. Their geometric nature makes them more robust to illumination variations. Now, the optimization is assessed by how well the associated edges align. The classic way to associate edges is to assign an edge pixel in one image to the nearest one in the second image. This simple association is prone to errors whenever edges are noisy or poorly repeatable. This motivated the integration of semantic constraints into this data association step: SNNF [195] matches an edge pixel in one image to the nearest one in the other image with the additional constraint that they must belong to the same semantic class.

When evaluated in an autonomous driving scenario using the vKITTI and the KITTI dataset, SNNF achieves lower trajectory error than Direct Sparse Odometry (DSO), ORB-SLAM2, and previous edge-based methods ANNF and ONNF. It also exhibits a larger convergence basin than existing edge-based VO, which means that it is more robust to

[6] https://vision.uvic.ca/image-matching-challenge/
[7] https://www.visuallocalization.net/workshop/cvpr/2020/

large camera displacement. The experiments also compare the edge repeatability of several semantic edge derivations. The edges derived by end-to-end learning methods are the most repeatable and lead to the best localization performance. While SNNF already improves the robustness of edge-based VO, one limit is the coarse edge localization. The learned edges exhibit a localization error that can go up to a few pixels. Future work will investigate reducing the noise in the edge localization to improve the localization final performance.

Semi-supervised Domain Adaptation for Semantics. The previous parts of this thesis showed how useful semantics are to make localization robust to image variations. Many trained segmentation models are already available online which facilitate the semantics' integration. However, these models usually exhibit poor generalization as soon as the image domain changes. This performance drop can go from segmentation noise to a full deterioration. To solve this issue, the standard approach is to finetune the model on a relatively small set of images from the target application but this requires supervision in the form of manually annotated images that can be costly and time-consuming to collect. This motivates the last part of this thesis to investigate two lighter supervisions to adapt semantic segmentation over domains.

Usually, a CNN generalizes well to images of which pixel distribution is similar to the one it was trained on. So **Chap. 7** proposes to leverage existing annotated datasets and transform them to bring their pixel distribution closer to the target data's one. Then, the CNN can be finetuned on the transformed data with the previous annotations. By training the network on images with a closer distribution to the target's one, this method improves the CNN's generalization. This approach draws motivation from the neural style-transfer that transforms the appearance of an image while preserving its content. Given a trained CNN, a source dataset already annotated, and a target set, the source images are transformed to exhibit the same CNN feature statistics as the target images. To do so, the source image is iteratively updated with a gradient computed by the backpropagation of two feature-based losses through the network and back to the image space. The resulting gradient is added to the image until convergence. The first loss constrains the image to keep the same content while the style loss forces it to adopt the target image's appearance.

This method is tested on a long-term monitoring dataset collected by local region partners. It is made of aerial images sampled every few years since 1950. They present a wide range of appearances from recent color images taken with multi-spectral cameras to analog images that were later digitized. Pixel-wise annotations have been collected

by a human for the 2015 images. Here, the goal is to transform the 2015 images to bring them closer to the 1950 distribution. The synthetic 1950 data is evaluated with the segmentation performance on the real 1950 data. The results show that such synthetic data achieves only a third of the performance reached with the real data. Another test trains the CNN on a black-and-white version of the 2015 dataset. The segmentation results are worse, which suggests that the style transformation does more than a simple color change. Even though the stylized data can not replace the real one, it proves useful to warm up the CNN. The network then needs fewer real examples and even reaches higher segmentation performances than without the warm-up.

The previous approach relied on the CNN feature property to represent the image's statistics. This information is brought back to the image space to transform the source data. Then the transformed data serves to finetune the CNN feature space. **Chap. 8** proposes to short-cut this process and only manipulate feature information. One drawback is that it requires semi-annotations in the form of approximately aligned images. The target CNN is initialized with the weights of the source CNN trained on the annotated data. Given a pair of source and target images, the target network is trained to generate the same feature maps on the target data as the source network does on the source image. This assumes that the images are approximately aligned, which naturally occurs in localization applications. The advantage is that there is no need to generate new data and finetuning becomes a simple regression problem. This method achieves similar performance as standard finetuning on the PASCAL VOC dataset with synthetic transformations. A qualitative study observed the images reconstructed from the target network's feature maps exhibit the same appearance as the images the network was trained on. This suggests that after the regression adaptation, the CNN project images with similar content and different appearance to the same point in the feature space. This is what is expected from a robust model.

9.2 Future work

This section lists the questions raised by this thesis that will be investigated in future work.

Local and Global Features. On the local detection from a trained CNN, the first challenge is to bring the detection to a sub-pixel precision for localization applications. A second issue is that only one feature map is used to compute the saliency map from

which keypoints are detected. This limits the information that ELF can leverage. Another limit is that the feature map used for detection is manually set. Qualitative experiments showed that the optimal choice may vary from one scene to another, and even over illuminations. So a numerical criterion for the automatic selection of the detection feature map must be defined.

The place recognition contributions in this thesis highlighted that edges are also a good source of local features. The literature usually discards edge features for they are too ambiguous to match. But the results achieved by SG-VLAD suggest that it is possible to leverage such information for feature-based localization. Current work has already started investigating such features and the qualitative results are encouraging even though defining a discriminative descriptor remains the main challenge.

Visual odometry. Integrating semantics into direct visual odometry has proven to make it more robust to illumination variations than existing approaches. One of the main challenges is the poor precision of semantic edge detection. It hinders the edge repeatability on which the optimization relies sensitively. One line of research currently investigated is to integrate feature-based geometric verification in the optimization loop, as a regularizer.

Unsupervised Domain Adaptation for Semantics. Semantics have played a pivotal role in most of the contributions of this thesis. To facilitate further research on semantic-based localization, future work will further investigate semi-supervised and unsupervised training methods to minimize the need for annotations.

Appendix A

NetVLAD Finetuning

A.1 NetVLAD Finetuning

This paragraph first recalls the NetVLAD training loss and parameters. The reader familiar with NetVLAD can skip this paragraph.

NetVLAD's input is a triplet of images $(\mathrm{I}, \mathrm{I}_p, \mathrm{I}_n)$ where I is the reference image, I_p a positive match image and I_n a hard negative image. A positive image is one that depicts the same scene as I and with similar viewpoint. I_n is the image with the nearest image descriptor but that does not overlap with the reference scene. This means that although this I_n has no similar content with the reference one, its image descriptor says that it does. NetVLAD is trained to generate the descriptors triplet (d, d_p, d_n) so that the descriptor distance $\|d - d_p\|_2$ between the reference and the positive images is small. And the descriptor distance between the reference image and the negative one $\|d - d_n\|_2$ is high. As there is no way to manually specific what 'high' and 'small' mean to a network, NetVLAD is trained so that the negative distance is at least higher than the positive one by a margin m: $m + \|d - d_p\|_2 \leq \|d - d_n\|_2$. The network is trained to minimize the loss $\mathscr{L} = \max(0, (m + \|d - d_p\|_2) - \|d - d_n\|_2)$. The right element is positive only when the negative descriptor distance is too small i.e. when the negative descriptor is too near to the reference image. Notice that the margin m is a key parameter to the training: if is too slow, it allows the negative descriptor to be near the reference one when it may be improved. And when it is too high, it leads to training instability as it requires the negative descriptor to exhibit values far from the reference one.

When training, triplets are sampled to improve the model efficiently by sampling the hardest examples. The hardest negative image is the one with the nearest descriptor to the reference one while it holds no overlap with the reference image. The positive example is simply the matching image with the nearest example. Authors take advantage of

the higher distribution of negative matches: at each training step, the model is training on one positive example N_{nh} hard negatives.

Notice that defining these triplets requires the camera ground-truth poses and that NetVLAD descriptors are available for all images. Obviously, it is not realistic to update the NetVLAD descriptors each time the network is optimized. Arandjelovic [9] propose to update the dataset descriptor every C training steps. Another computational bottleneck is the search for the hardest negative examples. One way to compute them is to find the N_{nh} hardest negative examples and update them every C training steps. The issue with this approach is that it requires to update the descriptors for the whole training set every C steps, and to search for the N_{nh} descriptors that are the furthest to query. Given that the training datasets usually encompass tens of thousands of images and that the descriptor computation requires one forward pass, this derivation requires too much time. Instead, the authors propose to update the N_{nh} hardest negative sample in the following way: sample randomly N_{nr} negatives noted B. The new N_{nh} hardest negative examples are the N_{nh} hardest examples in $A \cup B$. The parameter values are summarised in Tab. A.1.

Parameter	CMU-Seasons
batch size	3
N_{nr}	10
N_{nh}	3
margin m	1
C	1000
epochs	30
learning rate	10^{-4}
momentum	0.9
weight decay	10^{-3}
Learning rate decay	0.5
Decay epoch interval	5

Table A.1: NetVLAD finetuning parameters for CMU-Seasons and Symphony.

For CMU-Seasons, the park slices with ground-truth poses are {22-25} which are also the evaluation slices. Note that one constraint when training image retrieval systems is that the training data and the evaluation one must be spatially disjoint. To comply with this constraint, the model is finetuned on three of these slices and evaluated on the remaining one. One of 2 images is sampled from both the database and the query images of each slices. Two images match when the euclidean distance between their camera center is below $d_{pos} = 5m$. They are not matching when the distance is higher than

$d_{neg} = 10m$. This ensures that I_n has no overlap with the reference image I. Images withing a distance $d_{pos} < d < d_{neg}$ are not positives: they may hold some overlap with the reference image but not enough to require NetVLAD to generate similar descriptors. They are not negatives samples as the marginal overlap may lead to some similarity between the images' descriptors.

List of Figures

Bibliography

[1] Ign forests. http://professionnels.ign.fr/bdforet. 127

[2] Ign rpg. http://professionnels.ign.fr/rpg. 127

[3] Sandre. lenc12vlbenchmarks. 127

[4] Henrik Aanæs, Anders Lindbjerg Dahl, and Kim Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97(1):18–35, 2012. 90

[5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 90, 130

[6] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11075–11083, 2019. 48

[7] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2012. 26

[8] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *European Conference on Computer Vision*, pages 214–227. Springer, 2012. 26, 83, 88

[9] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2017. 7, 34, 44, 47, 52, 57, 153

[10] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 33

[11] Relja Arandjelović and Andrew Zisserman. Visual vocabulary with a semantic twist. In *Asian Conference on Computer Vision*, pages 178–195. Springer, 2014. 35

[12] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010. 111

[13] Hiwad Aziz. https://github.com/hiwad-aziz/kitti_deeplabxsxs, 2018. 117

[14] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–45, 2015. 34

[15] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015. 34

[16] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014. 33, 34, 44, 47, 54

[17] Hernan Badino, Daniel Huber, and Takeo Kanade. The CMU Visual Localization Data Set. http://3dvis.ri.cmu.edu/data-sets/localization, 2011. 51

[18] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 20, 128, 132

[19] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5173–5182, 2017. 83, 90, 99, 159

[20] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Britich Machine Vision Conference*. 28, 53, 84

[21] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006. 24, 83, 88

[22] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):509–522, 2002. 48, 67, 68, 70, 71, 146, 158

[23] Assia Benbihi, Stéphanie Arravechia, Matthieu Geist, and Cédric Pradalier. Image-based place recognition on bucolic environment across seasons from semantic edge description. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020. 6, 7, 8, 44, 106, 145

[24] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Elf: Embedded localisation of features in pre-trained CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7940–7949, 2019. 6, 8, 20, 27, 82, 147

[25] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Geometric and semantic visual words for scene recognition across seasons. *submitted*, 2019. 6, 7, 8, 106, 145

[26] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Learning sensor placement from demonstration for UAV networks. In *Proceedings of the IEEE Symposium on Computers and Communications*, 2019. 7, 30

[27] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Semi-supervised domain adaptation with representation learning for semantic segmentation across time. In *International Conference on Neural Information Processing*, pages 459–466. Springer, 2019. 6, 9, 133

[28] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006. 11

[29] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2013. 37

[30] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012. 129

[31] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *arXiv preprint arXiv:1604.03605*, 2016. 38

[32] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*, pages 778–792. Springer, 2010. 25, 83

[33] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986. 107

[34] John Canny. A computational approach to edge detection. In *Readings in computer vision*, pages 184–203. Elsevier, 1987. 116, 160

[35] Francine Catté, Pierre-Louis Lions, Jean-Michel Morel, and Tomeu Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical analysis*, 29(1):182–193, 1992. 26

[36] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 20, 128, 136, 137

[37] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 117

[38] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1800–1807, 2017. 90, 117

[39] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 539–546, 2005. 27

[40] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016. 28, 30, 84, 147

[41] GC-H Chuang and C-CJ Kuo. Wavelet descriptor of planar curves: Theory and applications. *IEEE Transactions on Image Processing*, 5(1):56–70, 1996. 45, 48, 49, 69

[42] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013. 25

[43] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of*

the IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016. 49

[44] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011. 23, 25

[45] Tung Dang, Christos Papachristos, and Kostas Alexis. Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2526–2533. IEEE, 2018. 38

[46] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 20, 28, 30, 84, 88, 147

[47] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, 2015. 111, 116, 160

[48] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. 2016. 139

[49] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 140

[50] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016. 20, 156

[51] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 20, 27, 28, 30, 84, 147

[52] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018. 18, 107, 112

[53] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 107

[54] Logan Engstrom. Fast style transfer. https://github.com/lengstrom/fast-style-transfer/, 2016. 130

[55] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 128, 136

[56] Bin Fan, Fuchao Wu, and Zhanyi Hu. Rotationally invariant descriptors using intensity order pooling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2031–2045, 2012. 28

[57] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*, 2014. 28, 30, 33, 81, 84, 147

[58] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 114, 116

[59] Dashan Gao, Sunhyoung Han, and Nuno Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):989–1005, 2009. 37

[60] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 40, 85, 124, 125, 126, 135, 161

[61] Abel Gawel, Carlo Del Don, Roland Siegwart, Juan Nieto, and Cesar Cadena. X-view: Graph-based semantic multi-view localization. *IEEE Robotics and Automation Letters*, 3(3):1687–1694, 2018. 37, 44

[62] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 111

[63] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 11

[64] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer, 2016. 34

[65] Gösta H Granlund. Fourier preprocessing for hand print character recognition. *IEEE Transactions on computers*, 100(2):195–201, 1972. 49

[66] Shane Griffith, Georges Chahine, and Cédric Pradalier. Symphony lake dataset. *The International Journal of Robotics Research*, 36(11):1151–1158, 2017. 7, 43, 44, 46, 52, 136

[67] Erdan Gu, Jingbin Wang, and Norman I Badler. Generating sequence of eye fixations using decision-theoretic attention model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 92–92, 2005. 38

[68] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1735–1742, 2006. 27, 84

[69] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015. 27, 84

[70] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 991–998, 2011. 136

[71] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 21, 24

[72] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 11, 12, 14, 15, 155, 156

[73] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 596–605, 2018. 27

[74] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 128

[75] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019. 119

[76] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 119

[77] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017. 140

[78] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 28

[79] Hervé Jégou and Ondřej Chum. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In *European conference on computer vision*, pages 774–787. Springer, 2012. 33, 72

[80] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pages 304–317. Springer, 2008. 53, 57

[81] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Packing bag-of-features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2357–2364, 2009. 32

[82] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010. 33, 44, 47, 72

[83] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2011. 33, 34, 44, 52, 53

[84] Tomas Jenicek and Ondrej Chum. No fear of the dark: Image retrieval under varying illumination conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9696–9704, 2019. 43

[85] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 111, 129

[86] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 126

[87] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, 2012. 38

[88] Fabjan Kallasi, Dario Lodi Rizzini, and Stefano Caselli. Fast keypoint features from laser scanner for robot localization and mapping. *IEEE Robotics and Automation Letters*, 1(1):176–183, 2016. 48

[89] Christopher Kanan, Mathew H Tong, Lingyun Zhang, and Garrison W Cottrell. Sun: Top-down saliency using natural statistics. *Visual cognition*, 17(6-7):979–1003, 2009. 38

[90] Jagat Narain Kapur, Prasanna K Sahoo, and Andrew KC Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3):273–285, 1985. 85, 87

[91] Michel Keller, Zetao Chen, Fabiola Maffra, Patrik Schmuck, and Margarita Chli. Learning deep descriptors with scale-aware triplet networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 28, 84

[92] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3251–3260, 2017. 35

[93] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 131

[94] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10. IEEE Computer Society, 2007. 107

[95] Laurent Kneip, Zhou Yi, and Hongdong Li. In *Britich Machine Vision Conference*. 109

[96] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011. 129

[97] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008. 38

[98] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 20, 34, 90, 131

[99] Srinivas SS Kruthiventi, Kumar Ayush, and R Venkatesh Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing*, 26(9):4446–4456, 2017. 38

[100] BG Kumar, Gustavo Carneiro, Ian Reid, et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5385–5394, 2016. 28

[101] Matthias Kummerer, Thomas SA Wallis, Leon A Gatys, and Matthias Bethge. Understanding low-and high-level contributions to fixation prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4789–4798, 2017. 20, 38, 156

[102] Pascal Lamblin and Yoshua Bengio. Important gains from supervised fine-tuning of deep architectures on large labeled sets. In *NIPS* 2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–8, 2010. 139

[103] Mans Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. A cross-season correspondence dataset for robust semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9532–9542, 2019. 48, 49, 57, 73

[104] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 31–41, 2019. 36

[105] K. Lenc, V. Gulshan, and A. Vedaldi. Vlbenchmkars. http://www.vlfeat.org/benchmarks/xsxs, 2011. 89

[106] Karel Lenc and Andrea Vedaldi. Large scale evaluation of local image feature detectors on homography datasets. In *Britich Machine Vision Conference*. 22

[107] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *European Conference on Computer Vision*, pages 100–117. Springer, 2016. 27

[108] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2548–2555, 2011. 26

[109] Konstantinos-Nektarios Lianos, Johannes L Schonberger, Marc Pollefeys, and Torsten Sattler. Vso: Visual semantic odometry. In *European Conference on Computer Vision*, pages 234–250, 2018. 36, 112

[110] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 30, 84, 131

[111] Tony Lindeberg. *Scale-Space Theory in Computer Vision*, volume 256. Springer Science & Business Media, 1993. 24

[112] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 128

[113] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014. 27, 28, 30, 33, 81, 147

[114] Justine Louvel and Vincent Gaudillat. *EUNIS, European Nature Information System, Système d'information européen sur la nature: classification des habitats: traduction française: habitats terrestres et d'eau douce*. MNHN, 2013. 127

[115] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1150–1157, 1999. 24

[116] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 21, 24, 46, 70, 83, 88

[117] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015. 40, 85, 135

[118] Elmar Mair, Gregory D Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *European conference on Computer Vision*, pages 183–196. Springer, 2010. 25

[119] Jonathan H Manton. Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650, 2002. 109

[120] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 416–423, 2001. 111

[121] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 25, 83, 88

[122] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 140

[123] Mayss' aa Merhy et al. *Reconnaissance de formes basée géodésiques et déformations locales de formes*. PhD thesis, Brest, 2017. 47

[124] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings of the IEEE International Conference on Computer Vision*, 2001. 24

[125] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 90

[126] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. 88, 89

[127] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017. 28

[128] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit. Learning to assign orientations to feature points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 107–116, 2016. 29

[129] Arsalan Mousavian, Jana Košecká, and Jyh-Ming Lien. Semantically guided location recognition for outdoors scenes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4882–4889. IEEE, 2015. 35

[130] Dibyendu Mukherjee, QM Jonathan Wu, and Guanghui Wang. A comparative experimental study of image feature detectors and descriptors. *Machine Vision and Applications*, 26(4):443–466, 2015. 22

[131] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 23, 32, 107, 112

[132] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2320–2327, 2011. 107

[133] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3465, 2017. 7, 34, 38, 43, 44, 47, 52, 53, 57, 84

[134] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 128

[135] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *arXiv preprint arXiv:1805.09662*, 2018. 20, 29, 84, 88, 90, 147

[136] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1717–1724, 2014. 139

[137] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier and Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. In *arXiv*, January 2017. 20

[138] Seonwook Park, Thomas Schöps, and Marc Pollefeys. Illumination change robustness in direct visual slam. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 4523–4530. IEEE, 2017. 107

[139] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. 26

[140] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 32

[141] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3384–3391, 2010. 32, 44, 46

[142] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 23, 32, 56

[143] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 32

[144] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2609–2616. IEEE, 2014. 107

[145] Cédric Pradalier and François Pomerleau. Multi-session lake-shore monitoring in visually challenging conditions. In *Field and Service Robotics*. Springer, 2019. 52

[146] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pages 3–20. Springer, 2016. 28, 35

[147] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 28

[148] Antoine Richard, Assia Benbihi, Cédric Pradalier, Vincent Perez, Philippe Durand, and Rosalinde Van Couwenberghe. Automated segmentation and classification of land use from overhead imagery. In *International Conference on Precision Agriculture*, 2018. 6, 8, 123, 124, 136

[149] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222, 2017. 132

[150] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic

segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 132

[151] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443. Springer, 2006. 25, 83

[152] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):105–119, 2010. 25

[153] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011. 25, 83, 88

[154] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 119

[155] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 119

[156] Paul-Edouard Sarlin, Frederic Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *Conference on Robot Learning*, pages 456–465, 2018. 84, 119

[157] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 7, 43, 44, 46, 51, 52, 62

[158] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 27, 84

[159] Fabian Schenk and Friedrich Fraundorfer. Robust edge-based visual odometry using machine-learned edges. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1297–1304. IEEE, 2017. 115

[160] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 23, 93

[161] Johannes L Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6959–6968, 2017. 22

[162] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6896–6906, 2018. 36

[163] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 40, 85

[164] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014. 28

[165] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, 2017. 41

[166] Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11651–11660, 2019. 27

[167] Oriane Siméoni, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Unsupervised object discovery for instance recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1745–1754. IEEE, 2018. 39

[168] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015. 27, 29, 84

[169] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 39, 40, 85, 139

[170] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1573–1585, 2014. 28

[171] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 20, 86, 90, 130, 137

[172] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, page 1470, 2003. 23, 32, 44, 46, 52, 53

[173] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 41, 85

[174] J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. Striving for simplicity: The all convolutional net. In *3rd International Conference on Learning Representations, Workshop Track Proceedings*, 2015. 40, 41, 85

[175] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6484–6490. IEEE, 2018. 36

[176] Hauke Strasdat, J Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2(3):7, 2010. 107

[177] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 92

[178] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017. 85

[179] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5109–5118, 2019. 34

[180] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 661–669, 2017. 28

[181] Gian Diego Tipaldi and Kai O Arras. Flirt-interest regions for 2d range data. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3616–3622. IEEE, 2010. 48

[182] Carl Toft, Carl Olsson, and Fredrik Kahl. Long-term 3d localization and pose from semantic labellings. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 650–659, 2017. 35, 36, 43, 47, 55, 57, 58, 65, 74, 75, 77, 106, 157, 158

[183] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *European Conference on Computer Vision*, pages 391–408. Springer, 2018. 36

[184] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010. 26

[185] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: aggregation across single and multiple images. *International Journal of Computer Vision*, 116(3):247–261, 2016. 33, 34

[186] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *4th International Conference on Learning Representations, Conference Track Proceedings*, 2016. 34

[187] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015. 33, 47

[188] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 883–890, 2013. 33, 47, 53, 78

[189] Tomasz Trzcinski, Mario Christoudias, Vincent Lepetit, and Pascal Fua. Learning image descriptors with the boosting-trick. In *Advances in neural information processing systems*, pages 269–277, 2012. 28

[190] Muhammad Usman, Abdul Manan Khan, Ahmad Ali, Sheraz Yaqub, Khalil Muhammad Zuhaib, Ji Yeong Lee, and Chang-Soo Han. An extensive approach to features detection and description for 2-d range data using active b-splines. *IEEE Robotics and Automation Letters*, 4(3):2934–2941, 2019. 48

[191] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5279–5288, 2015. 26, 83, 88, 89, 91

[192] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001. 24

[193] Joachim Weickert, Seiji Ishikawa, and Atsushi Imiya. Linear scale-space has first been proposed in japan. *Journal of Mathematical Imaging and Vision*, 10(3):237–252, 1999. 24

[194] Joachim Weickert, BM Ter Haar Romeny, and Max A Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998. 26

[195] Xiaolong Wu, Assia Benbihi, Antoine Richard, and Cédric Pradalier. Semantic nearest neighbor fields monocular edge visual-odometry. *arXiv preprint arXiv: 1904.00738*, 2019. 6, 8, 105, 148

[196] Xiaolong Wu and Cédric Pradalier. Illumination robust monocular direct visual odometry for outdoor environment mapping. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2392–2398. IEEE, 2019. 18, 112

[197] Xiaolong Wu and Cedric Pradalier. Robust semi-direct monocular visual odometry using edge and illumination-robust cost. *arXiv preprint arXiv:1909.11362*, 2019. 18

[198] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015. 111, 116, 160

[199] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016. 20, 29, 84, 88, 90, 95

[200] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, number CONF, 2018. 28

[201] Xin Yu, Sagar Chaturvedi, Chen Feng, Yuichi Taguchi, Teng-Yok Lee, Clinton Fernandes, and Srikumar Ramalingam. Vlase: Vehicle localization by aggregating semantic edges. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3196–3203. IEEE, 2018. 35, 43, 47, 57, 106

[202] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5964–5973, 2017. 35, 47, 48, 54, 110, 111, 116, 119, 140, 160

[203] Zhiding Yu, Weiyang Liu, Yang Zou, Chen Feng, Srikumar Ramalingam, BVK Vijaya Kumar, and Jan Kautz. Simultaneous edge alignment and learning. In *European Conference on Computer Vision*, pages 388–404, 2018. 48, 111, 116, 160

[204] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015. 27

[205] Charles T Zahn and Ralph Z Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on computers*, 100(3):269–281, 1972. 49

[206] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 40, 85

[207] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6325–6333, 2018. 27

[208] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32–32, 2008. 38

[209] Xu Zhang, X Yu Felix, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4605–4613, 2017. 27, 28

[210] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 20, 48, 49, 57

[211] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1224–1244, 2017. 31

[212] Naaman Zhou. Australian bushfires from the air: before and after images show scale of devastation. https://bit.ly/2G5JKci, 2019. 5, 155

[213] Yi Zhou, Laurent Kneip, and Hongdong Li. Semi-dense visual odometry for rgb-d cameras using approximate nearest neighbour fields. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6261–6268. IEEE, 2017. 18, 107, 111

[214] Yi Zhou, Hongdong Li, and Laurent Kneip. Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment. *IEEE Transactions on Robotics*, 35(1):184–199, 2019. 18, 111