Deep Multicameral Decoding for Localizing Unoccluded Object Instances from a Single RGB Image

Matthieu Grard, Emmanuel Dellandréa, Liming Chen

International Journal of Computer Vision, Special Issue on Deep Learning for Robotic Vision
Final authenticated version at: https://doi.org/10.1007/s11263-020-01323-0

Deep Multicameral Decoding for Localizing Unoccluded Object Instances from a Single RGB Image

Matthieu Grard · Emmanuel Dellandréa · Liming Chen

Received: 18 July 2018 / Accepted: 11 March 2020

Abstract Occlusion-aware instance-sensitive segmentation is a complex task generally split into region-based segmentations, by approximating instances as their bounding box. We address the showcase scenario of dense homogeneous layouts in which this approximation does not hold. In this scenario, outlining unoccluded instances by decoding a deep encoder becomes difficult, due to the translation invariance of convolutional layers and the lack of complexity in the decoder. We therefore propose a multicameral design composed of subtask-specific lightweight decoder and encoder-decoder units, coupled in cascade to encourage subtask-specific feature reuse and enforce a learning path within the decoding process. Furthermore, the state-of-the-art datasets for occlusion-aware instance segmentation contain real images with few instances and occlusions mostly due to objects occluding the background, unlike dense object layouts. We thus also introduce a synthetic dataset of dense homogeneous object layouts, namely Mikado, which extensibly contains more instances and inter-instance occlusions per image than these public datasets. Our extensive experiments on Mikado and public datasets show that ordinal multiscale units within the decoding process prove more effective than state-of-the-art design patterns for capturing position-sensitive representations. We also show that Mikado is plausible with respect to real-world problems, in the sense that it enables the learning of performance-enhancing representations transferable to real images, while drastically reducing the need of hand-made annotations for finetuning. The proposed dataset will be made publicly available.

Keywords Instance boundary and occlusion detection · Fully convolutional encoder-decoder networks · Synthetic data · Domain adaptation

Meaningful box proposals. Ambiguous box proposals.

Additional examples of dense object layouts in robotics.

Fig. 1: In dense object layouts, occlusions are mostly between instances that cannot be isolated in a rectangle. Mapping an image or a region that contains multiple similar instances to an instance-sensitive segmentation becomes ambiguous, thereby reducing the discriminative power of the encoded representations.

M. Grard
Siléane, 17 rue Descartes F-42000 Saint Étienne, France
Université de Lyon, CNRS, École Centrale de Lyon LIRIS UMR5205, F-69134 Lyon, France
Tel.: +33 (0)4 77 79 03 71
Fax: +33 (0)4 77 74 50 86
E-mail: m.grard@sileane.com

E. Dellandréa
Université de Lyon, CNRS, École Centrale de Lyon LIRIS UMR5205, F-69134 Lyon, France
E-mail: emmanuel.dellandrea@ec-lyon.fr

L. Chen
Université de Lyon, CNRS, École Centrale de Lyon LIRIS UMR5205, F-69134 Lyon, France
E-mail: liming.chen@ec-lyon.fr

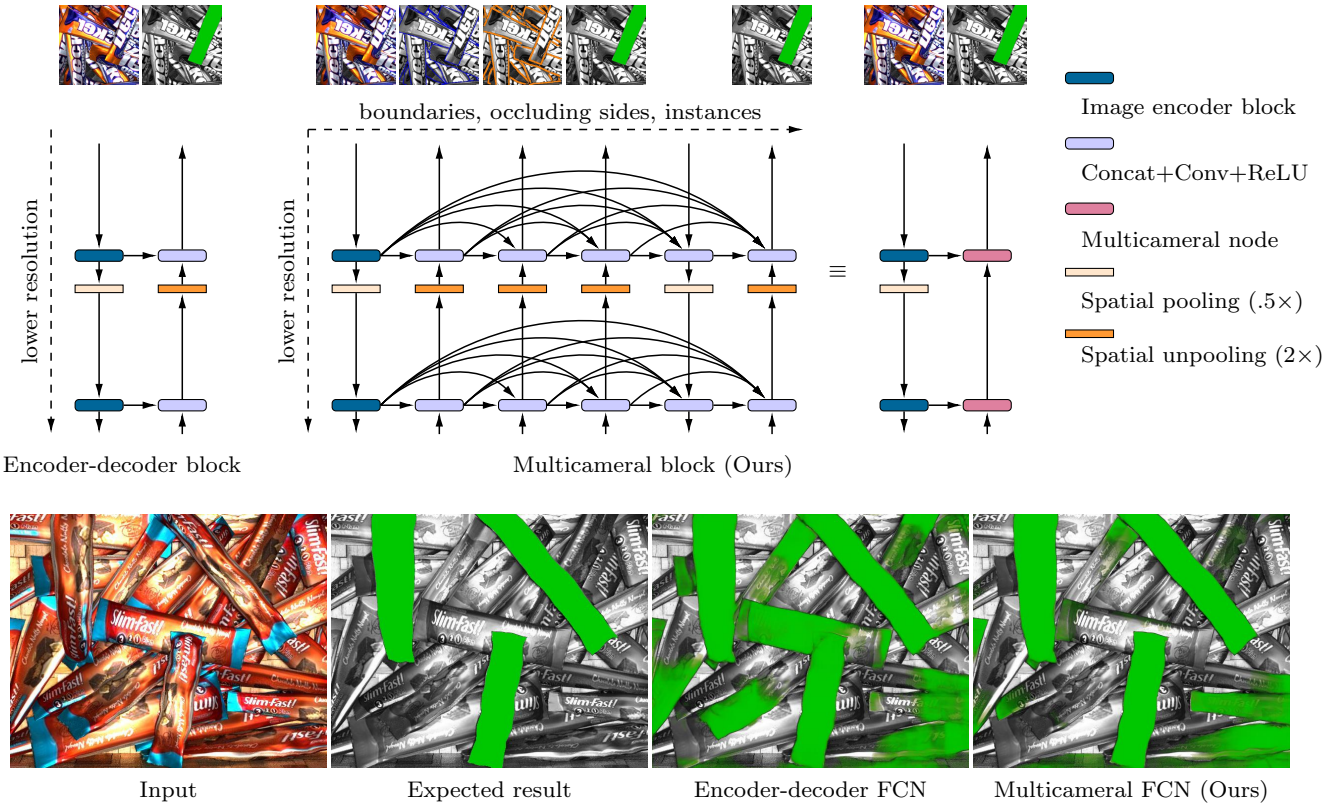


Fig. 2: Due to its built-in translation invariance, a deep encoder can hardly be decoded for distinguishing similar overlapping instances. We show the importance of decomposing the decoding process into ordinal subtasks to improve the attention to unoccluded instances in homogeneous layouts.

1 Introduction

Outlining object instances and understanding their spatial layout from a single RGB image without explicit object models is a core computer vision task in many robotic applications, such as object picking and autonomous driving in unknown environments. Indeed, the least occluded instances are often the most affordable ones to grasp or the closest obstacles to avoid. Automating such a task remains challenging as a robot must handle many variations of scene layouts from a mere grid of RGB values.

Deep fully convolutional networks (FCN) have become the state of the art for learning generalizable image representations due to their ability to capture multiscale invariants in trainable convolution kernels. In this context, a mainstream strategy for detecting salient instances consist in splitting the image segmentation into many region-wise segmentations. Specifically, a two-step FCN is trained to first isolate each instance in a bounding box by joint classification and regression of anchor boxes, then for each box proposal fire the pixels that belong to the visible and occluded instance parts (Follmann et al, 2019; Qi et al, 2019; Zhu et al, 2017) or to predefined affordance categories (Do et al, 2018). However, approximating an instance as a rectangle is not always relevant. Typically, in dense homogeneous layouts, many instances of the same object occlude each other. As a result, a box proposal often contains multiple instances (*c.f.* Figure 1).

In such object layouts, mapping an image or a region to an instance-sensitive segmentation becomes a difficult task, because a pixel-wise attention to specific instances requires position-dependent representations, whereas convolution kernels are translation invariant. Generally, pixel-wise labels are inferred by gradually combining low-resolution object-level semantics and higher-resolution local cues using a residual encoder-decoder (RED) network. In such a structure, the decoder aims to upsample the encoder latent representations. RED networks have proved efficient for inferring instance-agnostic categories (Chen et al, 2018) and instance boundaries (Deng et al, 2018; Ronneberger et al, 2015; Wang et al, 2017). However, a deep encoder can hardly be decoded for distinguishing similar overlapping instances, due to its built-in translation invariance (*c.f.* Figure 2). Most research efforts to improve object delineation have been put in the

encoder, using densely connected layers to deepen the encoder blocks (Huang et al, 2017), dilated convolutions to enlarge the receptive field at the lowest-resolution encoding level (Chen et al, 2018; Wang et al, 2018b; Yu and Koltun, 2016) or coordinate-aware convolutions to associate the latent representations with global pixel locations (Liu et al, 2018b; Novotný et al, 2018). These design patterns lead to low-resolution position-dependent representations of object categories, easier to be upsampled. However, in dense homogeneous layouts, the decoding process has greater importance because the diversity of objects to encode is much reduced while the pixel embeddings must discriminate between instances of the same object.

We therefore further the residual encoder-decoder design in order to approximate a mapping between single RGB images of homogeneous instance layouts and occlusion-aware instance-sensitive segmentations. Specifically, we propose a more complex decoding process to produce contextual pixel embeddings that better discriminate between similar instances. Our multicameral design consists of lightweight decoder and encoder-decoder units densely coupled in cascade, and differently supervised to decompose the complex task of outlining unoccluded instances into simpler ones: extracting image cues, detecting instance boundaries, detecting occluding boundary sides, firing the pixels of unoccluded instances, refining the segmentation. In contrast with the state-of-the-art design patterns for capturing position-dependent representations, our approach encourages subtask-specific feature reuse and longer-range relations within the decoding process, thus improving the attention to unoccluded instances in homogeneous layouts (*c.f.* Figure 2).

Furthermore, the state-of-the-art datasets for joint instance delineation and occlusion detection (Follmann et al, 2019; Fu et al, 2016; Qi et al, 2019; Wang and Yuille, 2016; Zhu et al, 2017) are intrinsically designed for the foreground/background paradigm. As shown by Figure 3, the images in these datasets contain few instances and a large number of occlusions are due to objects occluding the background. In addition, these datasets suffer from biased data distributions due to limited variations and error-prone hand-made annotations. They can hardly be extended, as producing a pixel-wise ground truth for instance boundaries and occlusions is a tedious and time-consuming task for human annotators. Specifically, these datasets never showcase homogeneous layouts with many occlusions between instances, although it is a common scenario in robotic applications for manufactured object manipulation.

Therefore, we also propose a synthetic dataset of dense homogeneous layouts for evaluating the learning of an instance-sensitive mapping, through the canonical scenario of many sachets piled up in bulk. Our data generation pipeline flexibly enables lots of inter-instance occlusion variations and error-free annotations, unlike datasets of real images.

In summary, our contribution is two-fold:

- A *multicameral* FCN design to approximate a more complex decoding function for dense homogeneous layouts. Our extensive experiments show that introducing complexity and task decomposition into ordinal subtasks within the decoding process proves more effective than the state-of-the-art design patterns for capturing position-dependent representations, thus improving the attention to unoccluded instances from a single RGB image.
- A simulation-based pipeline, referred to as *Mikado*, to evaluate the proposed model on dense homogeneous instance layouts. Our synthetic data[1] extensibly contains more occlusions between similar instances than the public datasets for occlusion-aware instance segmentation. We show that the proposed data is plausible with respect to real-world problems, through experiments on transfer learning from Mikado to D2SA, a public dataset of real-world heterogeneous object layouts (Follmann et al, 2019).

Our paper is organized as follows. After reviewing the related work in Section 2, we describe the proposed model in Section 3, the proposed dataset in Section 4, then our experimental protocol in Section 5. Our results are finally discussed in Section 6.

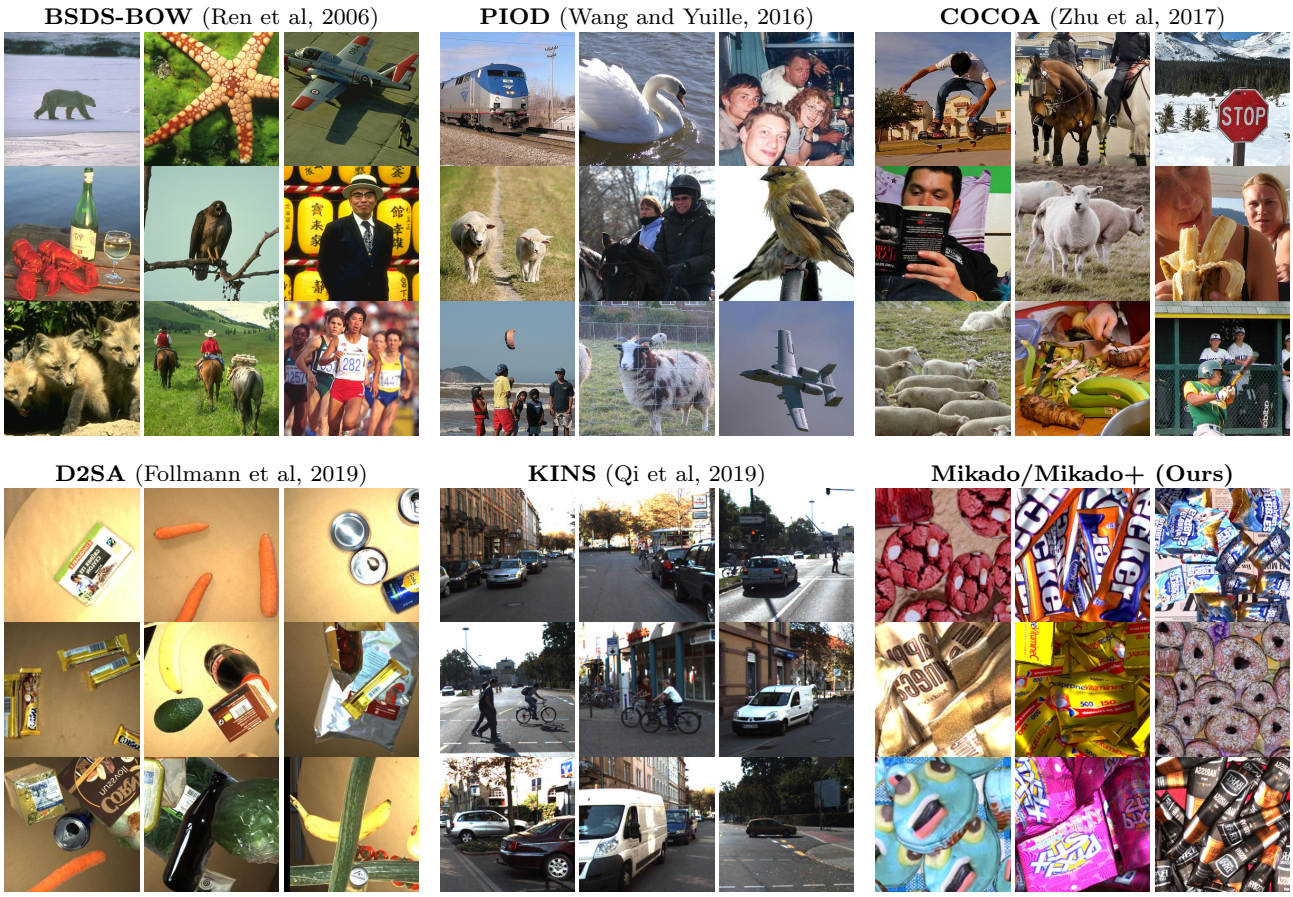
2 Related Work

Occlusion-aware instance-wise attention lies at the intersection of salient instance segmentation and occlusion detection. Also, the proposed multicameral design is composed of shared or task-specific encoders and decoders. In this section, we thus review the state of the art on salient instance segmentation and occlusion detection from a single RGB image, FCN architectures for pixel multi-labeling, and the public datasets for joint instance segmentation and occlusion detection.

2.1 Salient Instance Segmentation

Graph-based segmentation Instance delineation has been approached further to pixel-wise object categorization. Specifically, an instance-agnostic category is first assigned to each pixel, then the pixels within each category

[1] Publicly available at https://mikado.liris.cnrs.fr



Dataset	Average image size	Number of images	Number of instances	Instances per image	Inter-instance occlusions per image	Background pixels per image	Ground-truth annotations
BSDS-BOW[1]	432×369	200	–	–	–	–	Human-made
PIOD	469×386	10,100	24,797	2.5	1.3	69%	
COCOA[2]	578×483	3,823	34,884	9.1	13.5	33%	
D2SA[2]	1962×1569	5,600	28,703	5.1	2.8	79%	
KINS	1695×362	14,991	187,730	12.5	8.0	92%	
Mikado (Ours)	640×512	2,400	48,184	**20.1**	**52.9**	**24%**	**Computer-generated**
Mikado+[3] (Ours)	640×512	14,560	459,002	**31.5**	**60.5**	**24%**	

[1] The empty cells are due to the ground truth that consists only of object part-level oriented edges.
[2] The statistics are only on the train and validation subsets as the test subset is not provided.
[3] Mikado+ is an extension of Mikado used only to show the impact of a richer synthetic data distribution.

Fig. 3: State-of-the-art datasets for occlusion-aware boundary detection (BSDS-BOW, PIOD) and amodal instance segmentation (COCOA, D2SA, KINS) compared with our synthetic dataset. Unlike the state-of-the-art datasets in which occlusions are mostly due to objects occluding the background, Mikado contains more instances and occlusions between instances per image, thus better representing the variety of occlusions.

region are grouped into instances using graphical models, such as watershed transforms from inferred energy maps (Bai and Urtasun, 2017) or superpixel-based proposals (Kirillov et al, 2017; Li et al, 2017; Pont-Tuset et al, 2017). Indeed, in scenes with few similar or many heterogeneous instances, category masks effectively reduce the search space and partially reveal instance boundaries, as category boundaries are also instance boundaries. However, in scenes full of many instances of the same class (Figure 1), such a categorization is of little use. Defining instead instance-sensitive categories also fails, due to the built-in translation invariance of FCNs (Figure 2).

Recurrent segmentation Instance segmentation has also been formulated as a recurrent process (Kong and Fowlkes, 2018; Ren and Zemel, 2017; Romera-Paredes and Torr, 2016). Specifically, a recurrent FCN is trained to iteratively update a mean-shift clustering (Kong and Fowlkes, 2018) or iteratively outline each instance (Ren and Zemel, 2017; Romera-Paredes and Torr, 2016). Such memory-based pipelines are nevertheless harder to train than feedforward networks. (Ren and Zemel, 2017; Romera-Paredes and Torr, 2016) also assume a stationary scene, wheras in robotic applications, the scene is likely to change between two iterations due to physical interactions with the detected instances.

Proposal-based segmentation Alternatively, state-of-the-art strategies rely on two-step FCNs trained to first isolate each instance in a rectangle, then infer the corresponding mask after pooling the high-level features in the box proposal (Dai et al, 2016; Fan et al, 2019; Hayder et al, 2017; He et al, 2017; Liu et al, 2018c). Although these approaches are good at producing connected pixel clusters, the resulting mask boundaries suffer from the pooling quantization effect. Starting instead from binary rectangle masks on the box detector's last feature map (Fan et al, 2019) or using a distance transform (Hayder et al, 2017) to infer instance masks improves instance delineation, but still for instances that can fit a rectangle. As discussed in our introduction, these approaches also poorly address the problem of translation variance using FCNs, particularly in the case of multiple overlapping instances of the same object. Interestingly, mixing convolutional embeddings with hard-coded non-convolutional information, such as pixel locations, enables improvements in distinguishing adjacent instances (Liu et al, 2018b; Novotný et al, 2018).

2.2 Occlusion Detection

Depth estimation Finding occlusion relations has mostly been studied jointly with depth estimation in multiview contexts (Geiger et al, 1995; Grammalidis and Strintzis, 1998; Zitnick and Kanade, 2000) and motion sequences (Ayvaci et al, 2010, 2012; He and Yuille, 2010; Humayun et al, 2011; Stein and Hebert, 2006; Sun et al, 2014; Williams et al, 2011), as occlusions often translate into missing pixel correspondences in different points of view or consecutive frames. Recent works have more ambitiously focused on learning-based monocular 3D reconstruction using FCNs (Eigen et al, 2014; Fu et al, 2018; Gan et al, 2018; Li et al, 2015; Liu et al, 2016), but the results are still less accurate than standard multiview 3D reconstruction algorithms, and these techniques require sensor-specific ground-truth depth maps difficult to obtain. Although depth estimation brings relevant hints such as depth discontinuities, understanding occlusions is possible without putting effort into an explicit dense 3D reconstruction, as shown hereinafter.

Amodal/multiclass segmentation In keeping with box proposal-based instance segmentation (He et al, 2017; Liu et al, 2018c), two-step FCNs have been adapted for inferring, in each box proposal, either the mask including the visible and occluded instance parts (Follmann et al, 2019; Qi et al, 2019; Zhu et al, 2017) or a multiclass segmentation according to predefined affordance categories (Do et al, 2018). However, in addition to the cons of box proposal-based segmentation, inferring masks including occluded instance parts, referred to as *amodal segmentation*, is ambiguous because some pixels are attached to something invisible, whereas these pixels visually belong to another instance. Without explicit object models, the learning process is then conditioned on a guess only from global pixel relations, while fine-grained inferences require local pixel relations as well. Amodal annotations are also difficult to obtain unless synthesizing training images, leading to a domain shift. Defining instead affordance categories seems more reasonable, but in (Do et al, 2018), affordances are implicitly mapped to object part categories. For example, wrapping grasp affordances are cylinder-like objects such as bottles, bowls, knife handles. In a scene full of overlapping instances of the same affordance category, this strategy is prone to fail.

Oriented boundary detection FCNs prove more suitable for learning oriented contours, as this pixel labeling task does not require translation variance. Specifically, state-of-the-art approaches employ encoder-decoder networks including two task-specific decoders for recovering instance boundaries and occlusion-based orientations respectively (Wang et al, 2018a; Wang and Yuille, 2016). However, these approaches have two drawbacks. First, occlusions are modelled as pixel-specific raw orientations specifying the occlusion relations, without guarantee of continuity. As a consequence, a post-inference step is needed to adjust the noisy inferred orientations using the local tangent vectors of the inferred boundaries. Most importantly, the inferred boundaries are not guaranteed to be closed. As a consequence, instance masks cannot be easily extrapolated, *e.g.* by considering the dual connected components. An iterative refinement procedure has been proposed (Batra et al, 2019), but does not really solve the issue.

2.3 Pixel Multi-labeling

Encoder-decoder networks First introduced for single-task setups, such as semantic segmentation (Badrinarayanan et al, 2017) and instance boundary detection (Yang et al, 2016), encoder-decoder networks are designed to infer pixel labels despite the spatial resolution loss when encoding object-level semantics. Specifically, the encoder produces deep hierarchical features, then the decoder gradually outputs a probability map using symmetric unpooling stages (*c.f.* Figure 4a). However, in a sequential encoder-decoder, the pixel labels are inferred only from the last encoder feature maps, where the information is the most spatially compressed. Instead, a multiscale view can be given to the decoder through holistically-nested connections (Figure 4b) (Liu et al, 2017; Maninis et al, 2016; Xie and Tu, 2015). Nevertheless, such a late fusion requires to upsample all the latent representations to the image resolution. A progressive multiscale decoding through scale-specific skip connections between the encoder and decoder (*c.f.* Figure 4c) has consequently proved superior (Deng et al, 2018; Ronneberger et al, 2015; Wang et al, 2017). Indeed, at each decoding stage, the lower-resolution but higher-level semantics are merged with the higher-resolution information lost after pooling the encoder features of the current scale. Note that in application contexts requiring high resolutions, residual encoder-decoder networks may suffer from checkerboard artifacts, also referred to as the gridding effect (Guan et al, 2018; Liu et al, 2018a; Shi et al, 2016). Interestingly, coupling residual encoder-decoder networks via cross-network skip connections helps to refine the localization of visual landmarks (Tang et al, 2018).

Multi-task learning Sharing representations in learning multiple tasks generally enables to capture more generalizable invariants. In the context of semantic segmentation, (Luo et al, 2017) proposed to merge local and global semantics through a dual-task training, by jointly decoding pixel labels and inferring image labels. Image-level classification is however unfeasible in a category-agnostic problem, although detecting instance boundaries and inter-instance occlusions require global cues as well. For pixel multi-labeling, various strategies of knowledge sharing have been explored, such as progressive layer splitting (Misra et al, 2016), dynamic task loss weighing (Kendall et al, 2018), skip connection-like attention masks between a shared network and task-specific ones (Liu et al, 2019). These works are however focused on best learning task-shared and task-specific features to excel in every task. In this work, we are rather interested in exploiting an ordinal task decomposition to enforce a learning path, but not to excel in every subtask.

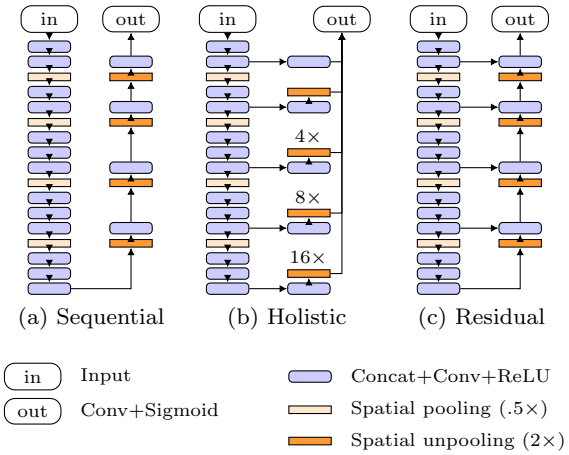


Fig. 4: State-of-the-art decoding strategies for boundary detection, using a VGG16-based (Simonyan and Zisserman, 2015) encoder. Best viewed in color.

2.4 Datasets

Oriented boundary detection Monocular occlusion-aware boundary detection raised interest with the BSDS Border Ownership dataset (BSDS-BOW) (Ren et al, 2006), which contains 200 real images from the BSDS500 dataset (Martin et al, 2001), manually annotated with object part-level oriented contours. As state-of-the-art FCNs require more training data, (Wang and Yuille, 2016) presented the PASCAL Instance Occlusion Dataset (PIOD), consisting of 10,100 manually annotated real images from the PASCAL VOC Segmentation dataset (Everingham et al, 2015). Despite their challenging intra-class variability, the images contain few instances and inter-instance occlusions (*c.f.* Figure 3).

Amodal segmentation (Follmann et al, 2019; Qi et al, 2019; Zhu et al, 2017) also released datasets of real images, respectively the KITTI INStance dataset (KINS), the Densely Segmented Supermarket Amodal dataset (D2SA) and the COCO Amodal dataset (COCOA), that are subsets of larger datasets for box proposal-based instance segmentation, respectively KITTI (Geiger et al, 2013), COCO (Lin et al, 2014) and D2S (Follmann et al, 2018), manually augmented with ground-truth amodal annotations. However, overcrowded scenes are also not represented in these datasets. Moreover, the ground-truth amodal annotations result from guesses, thereby introducing human biases in the learning process.

Synthetic images Synthetic datasets have emerged in various contexts as they offer rich multimodal annotations from fully controlled environments (Brégier et al, 2017; Gaidon et al, 2016; Grard et al, 2018; McCormac et al, 2017; Ros et al, 2016). Yet, in these datasets, dense homogeneous layouts have received little attention. Proposed for evaluating pose detection and estimation, the Siléane dataset (Brégier et al, 2017) consists of top-view depth images of identical rigid instances in piles. Similarly, (Grard et al, 2018) suggested synthetic depth maps of scanned objects instantiated in bulk. These synthetic datasets are however generated only for depth-based perception and elude the learning from a single RGB image.

3 Proposed Model

In this section, we first describe the proposed multicameral structuring for occlusion-aware instance-wise attention. Second, we detail the associated loss function.

3.1 Problem Statement

We aim to approximate a mapping between RGB images and instance-sensitive segmentations. As a showcase scenario, we look for sets of non-overlapping connected pixel clusters that represent unoccluded instances (see Figure 2). Formally, let $\mathcal{X}$ be our set of $|\mathcal{X}| \in \mathbb{N}^\star$ RGB images, and $\mathcal{P}$ the set of pixel locations. For an image of width $W \in \mathbb{N}^\star$ and height $H \in \mathbb{N}^\star$, we write $P = W \times H$, and $\mathcal{P} = \{1, ..., W\} \times \{1, ..., H\}$. We aim at approximating a function f defined as follows:

$$f \colon \mathcal{X} \to \{0,1\}^P, \; X \mapsto Y. \tag{1}$$

Given an image $X^n \in \mathcal{X}$, a pixel $\mathbf{p} \in \mathcal{P}$ is fired, *i.e.* $Y^n_{\mathbf{p}} = 1$ if it belongs to an unoccluded instance.

3.2 Proposed Architecture

Generally, a residual encoder-decoder (RED) network is a sequence of scale-specific encoding feature transforms E_s, and residual decoding feature transforms D_s such that:

$$\mathbf{x}_s = E_s(\mathbf{x}_{s-1}), \tag{2}$$

$$\mathbf{y}_s = D_s(\mathbf{y}_{s+1}, \mathbf{x}_s), \tag{3}$$

where $\mathbf{x}_s$ and $\mathbf{y}_s$ are the latent image representations at the resolution level s in the encoder and decoder respectively. For example, $\mathbf{x}_1 = E_1(X)$. If we note $E = \{E_s\}_{s\in\{1,...,S\}}$ and $D = \{D_s\}_{s\in\{1,...,S\}}$ then a RED network is a sequence $[E, D]$. In a RED network, the decoder aims to gradually upsample the deep representations of the encoder. This is however unsufficient to discriminate between instances of the same object.

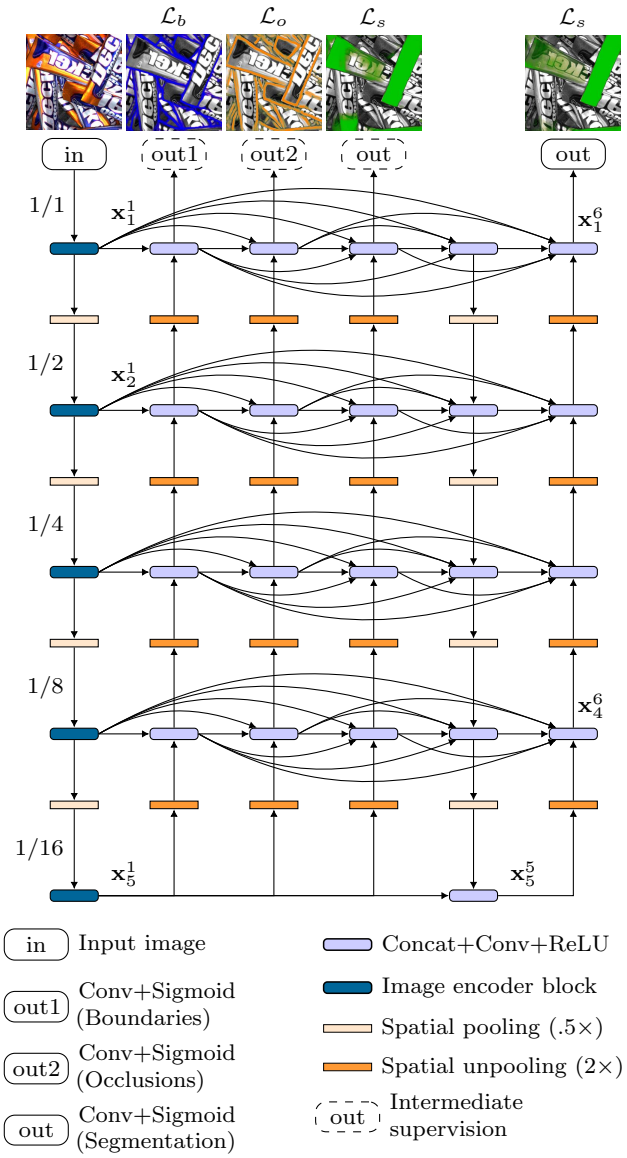


Fig. 5: Proposed multicameral structuring with ordinal intermediate supervisions (MC6†) for monocular attention to unoccluded instances. Best viewed in color.

By contrast, a multicameral (MC) network is a sequence of T residual decoder and encoder-decoder units, densely connected through resolution-wise skip connections, to approximate a more complex decoding function (see Figure 5). If we define encoders and decoders as multiscale feature transforms, then a multicameral structuring is a matrix-like layout of latent representations at S different resolutions. Each row thereby conveys high-level semantics at a fixed resolution. As

the starting point is an image, the first element is a deep encoder based on a common backbone, for example a VGG16 encoder (Simonyan and Zisserman, 2015). The first three decoders in cascade gradually recover the instance boundaries, the occluding boundary sides, and the segmentation outlining the unoccluded instances respectively. These ordinal units aim to structure the decoding process. It also encourages subtask-specific feature reuse: an occluding boundary side is expected to be near an instance boundary, and a pixel in an unoccluded instance is expected to be isotropically surrounded by occluding boundary sides. After these decoders, an encoder-decoder unit refines the segmentation.

Formally, let $\mathbf{x}_s^t$ be the latent representation at the row $s \in \{1, ..., S\}$ and column $t \in \{1, ..., T\}$. Then an encoding transform E_s^t and a decoding transform D_s^t at this position are defined respectively as:

$$\mathbf{x}_s^t = E_s^t(\mathbf{x}_{s-1}^t, \mathbf{x}_s^{t-1}, ..., \mathbf{x}_s^1), \tag{4}$$

$$\mathbf{x}_s^t = D_s^t(\mathbf{x}_{s+1}^t, \mathbf{x}_s^{t-1}, ..., \mathbf{x}_s^1). \tag{5}$$

If we note $E^t = \{E_s^t\}_{s\in\{1,...,S\}}$ and $D^t = \{D_s^t\}_{s\in\{1,...,S\}}$, then a multicameral design is the sequence $[E^1, D^2, D^3, D^4, E^5, D^6]$. In the following, we refer to a multicameral structure of T columns as MCT. For examples, MC4 $= [E^1, D^2, D^3, D^4]$, MC3 $= [E^1, D^2, D^3]$, and RED $=$ MC2 $= [E^1, D^1]$.

Feature transforms In the decoder and encoder-decoder units except the first encoder, the default encoding and decoding feature transforms consist of three operations: (1) concatenate the inputs along the channel axis (Concat); (2) apply a pixel-wise affine transformation (Conv); (3) apply a non-linear activation (ReLU). Only the transforms E_s^1 in the first encoder consists of more operations, such as sequential convolutions, to match common encoder backbones, such as a VGG16-based encoder (Simonyan and Zisserman, 2015). The encoder and decoder transforms $E_s^{t>1}$ and $D_s^{t>1}$ of a row s have the same number of filters. In practice, we set this number to be half the number of layers of the encoder representation (see details in our experimental setup in Section 5). In our experiments, we also consider the sparse use of alternative feature transforms for capturing position-dependent representations (*c.f.* Figure 6 for an overview of these transforms).

Skip connections We use skip connections by concatenation. Concatenation is favored over element-wise max or sum operators because such operators are special cases of concatenation. Formally, let $K \in \mathbb{N}^\star$ be the depth of two layers to merge, and $e, d, f \in \mathbb{R}^K$ feature vectors respectively for the encoder, the decoder, and the resulting fusion. Let $w, w' \in \mathbb{R}^{K\times K}$ be trainable parameters. Using element-wise max operators: $\forall k \in \{1, ..., K\}$, $f_k = \sum_{i=1}^K w_{ik} \max(e_{ik}, d_{ik})$. Using element-wise sum operators: $\forall k \in \{1, ..., K\}, f_k = \sum_{i=1}^K w_{ik}(e_{ik} + d_{ik})$. Using concatenation, $\forall k \in \{1, ..., K\}, f_k = \sum_{i=1}^N (w_{ik}e_{ik} + w'_{ik}d_{ik})$. If needed, an element-wise sum operator can then be modelled by setting $w = w'$. Similarly, an element-wise max operator can be obtained by setting $w_{ik} = 0$ or $w'_{ik} = 0$ depending on which of the ith encoder or decoder channel has greater importance.

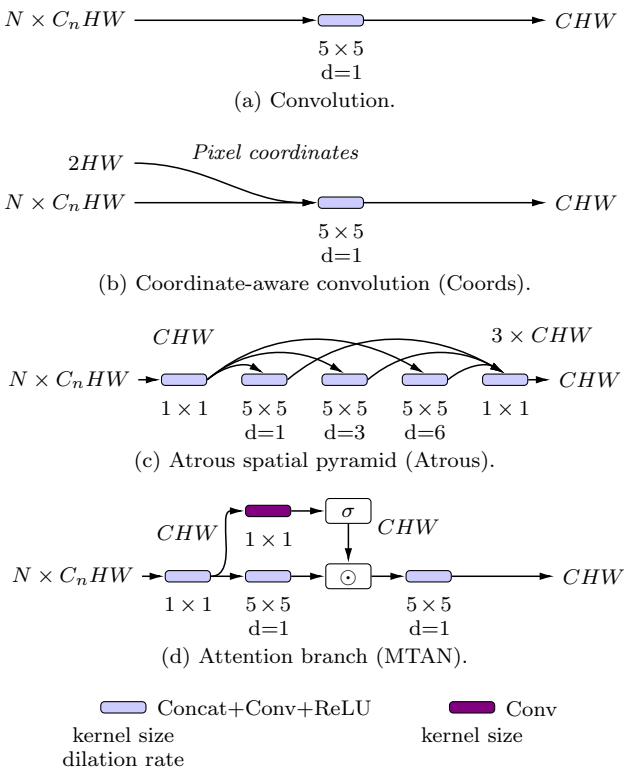


Fig. 6: State-of-the-art node-level mechanisms for learning a contextual representation of size CHW from N latent representations of size C_nHW respectively, where $n \in \{1, ..., N\}$.(a) Soft feature sampling using gradient-based weights. (b) Features are attached to global pixel coordinates before sampling (Liu et al, 2018b; Novotný et al, 2018). (c) Longer-range sampling using aggregated dilated convolutions (Chen et al, 2018; Wang et al, 2018b; Yu and Koltun, 2016). (d) Soft feature sampling using inferred masks (Liu et al, 2019).

Pooling types We use max operators in our spatial pooling layers, except in in the encoder (E^5) for refinement. In E^5, we use instead average pooling to gradually average the pixel embeddings within each instance. As a consequence, if the decoder D^4 infers an instance part instead of the whole instance, the representation of this instance will be altered. However, if an entire instance

is correctly classified, then its average pixel embedding will remain unchanged. This behavior would not be possible with max pooling because max operators highlight salient pixel embeddings. A wrongly classified instance part could then represent the whole instance.

3.3 Proposed Training

A multicameral structure is an acyclic graph, trainable end-to-end. As detecting instance boundaries, detecting occluding boundary sides, and outlining unoccluded instances can be formulated as binary classification tasks, we use balanced cross-entropy loss functions, with instance boundary-aware penalties to synchronize the different supervisions. We are aware of alternative loss functions that address the imbalance between positive and negative examples (Deng et al, 2018; Lin et al, 2017; Yu et al, 2018). As it is not our main focus in this work, we leave the reader to adapt the following loss functions if needed.

Loss functions Formally, let $\mathbf{p} \in \mathcal{P}$ be a pixel location – typically $\mathcal{P} = \{1, .., W\} \times \{1, .., H\}$ for an image of width $W \in \mathbb{N}^*$ and height $H \in \mathbb{N}^*$. We note $\mathcal{N} = \{1, .., N\}$ where $N \in \mathbb{N}^*$ is the number of training images, and $M_\mathbf{p} \in \mathcal{V}$ the value at location $\mathbf{p} \in \mathcal{P}$ in a matrix $M \in \mathcal{V}^\mathcal{P}$. Let $B^n, O^n, Y^n \in \{0, 1\}^P$ be the ground-truth binary images for instance boundaries, occluding boundary sides, and segmentation respectively. Let $\hat{B}^n$, $\hat{O}^n$, $\hat{Y}^n \in [0, 1]^P$ be the corresponding network inferences.

– For instance boundary detection, the decoder D^2 minimizes the loss function $\mathcal{L}_b(\theta)$ defined as follows:

$$\mathcal{L}_b(\theta) = -\frac{1}{|\mathcal{N}||\mathcal{P}|} \sum_{n \in \mathcal{N}} \sum_{\mathbf{p} \in \mathcal{P}} \alpha B^n_\mathbf{p} \log(\hat{B}^n_\mathbf{p}) + (1 - B^n_\mathbf{p}) \log(1 - \hat{B}^n_\mathbf{p}), \quad (6)$$

where $\alpha \in \mathbb{R}$ is a penalty to counterbalance the low number of boundary pixels against non-boundary pixels. In our experiments, we set $\alpha = 10$.

– For occluding boundary side detection, the decoder D^3 minimizes the loss function $\mathcal{L}_b(\theta)$ defined as follows:

$$\mathcal{L}_o(\theta) = -\frac{1}{|\mathcal{N}||\mathcal{P}|} \sum_{n \in \mathcal{N}} \sum_{\mathbf{p} \in \mathcal{P}} \alpha O^n_\mathbf{p} \log(\hat{O}^n_\mathbf{p}) + \beta(1 - O^n_\mathbf{p}) \log(1 - \hat{O}^n_\mathbf{p}), \quad (7)$$

where $\beta = \alpha \; if \; B^n_\mathbf{p} = 1 \; else \; 1$.

– For segmentation, the decoders D^4 and D^6 both minimize the loss function $\mathcal{L}_s(\theta)$ defined as follows:

$$\mathcal{L}_s(\theta) = -\frac{1}{|\mathcal{N}||\mathcal{P}|} \sum_{n \in \mathcal{N}} \sum_{\mathbf{p} \in \mathcal{P}} \alpha Y^n_\mathbf{p} \log(\hat{Y}^n_\mathbf{p}) + \beta(1 - Y^n_\mathbf{p}) \log(1 - \hat{Y}^n_\mathbf{p}))). \quad (8)$$

In the following, if a multicameral structure MCT is trained with these ordinal intermediate supervisions, we write MCT†. For example, MC3† is a bicameral structure trained for occlusion-aware boundary detection. RED=MC2=MC2† is a residual encoder-decoder network trained for segmentation.

Ground truth generation For each training and test images, we assume that we have the corresponding instance segmentation and the corresponding depth or instance-wise order (in that case, we consider it as a pseudo-depth). The depth (or pseudo-depth) is only used to create the ground truth, but never as input modality.

– The ground-truth boundaries are trivially derived from the instance segmentation.
– For generating the ground-truth occluding boundary sides, we sweep all the ground-truth instance boundaries and at each boundary pixel, we binarize the centered local region by computing the mean Z-offset in each segment of the region (see Figure 16 in appendix). In the end, the ground truth for occlusions is a binary image in which the positive pixels are the instance boundaries slightly translated to one side or another, according to the relative depth difference of the boundary sides. Note that local patches that contain more than two segments are fully set to 0 as they cannot be binarized. This proves to be a reasonable limitation as in practice an overwhelming majority of boundary pixels are between only two instances or between an instance and the background (*e.g.*, 97.1% of the boundary pixels in Mikado, and 99.4% in PIOD). We leave for future work the study of the minority of pixels at the junction of more than two instances.
– For generating the ground-truth segmentation outlining the unoccluded instances, we compute the number of occluding boundary pixels within each instance. If this ratio is very close to the instance perimeter, then the instance is considered as unoccluded.

4 Proposed Dataset

In this section, we describe the proposed pipeline for generating synthetic homogeneous instance layouts, referred to as Mikado.

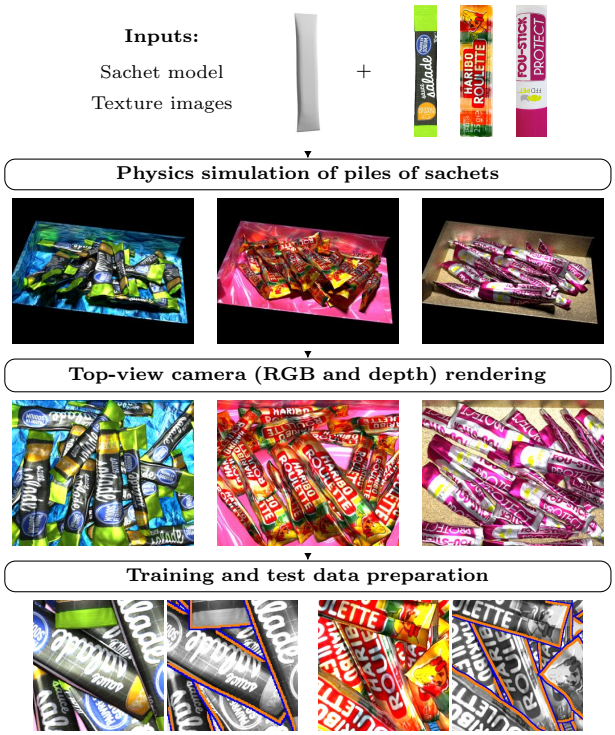


Fig. 7: Overview of the Mikado pipeline (best viewed in color). Given a mesh template and texture images, piles of deformed instances are generated using a physics engine. A top-view camera is then rendered to capture RGB and depth. The synthetic images and their annotations (ground-truth boundaries are in blue, unoccluded side in orange) are finally prepared to be fed-forward through the network.

4.1 Data Generation

In the same vein of (Brégier et al, 2017; Grard et al, 2018), we generate synthetic data using custom code on top of Blender (Blender Online Community, 2016) by simulating scenes of objects piled up in bulk and rendering the corresponding top views, as depicted in Figure 7. More precisely, after modelling a static open box and, on top, a perspective camera, a variable number of object instances, in random initial pose, are successively dropped above the box using Blender's physics engine (a video showing the generation of a scene is provided in supplementary material). We then render the camera view, and the corresponding depth image, using Cycles render engine. In this configuration, we ensure a large pose variability and a lot occlusions between instances. The ground-truth unoccluded instances and occluding instance boundary sides can be trivially derived from depth (*c.f.* Figure 16).

However, differently from (Brégier et al, 2017; Grard et al, 2018), we consider here piles of many instances with intra-class variations and using only RGB as input modality. We generate RGB images of sachets piled up in bulk by randomly applying global and local deformations to one mesh template of sachet that we texture successively with one out of 120 texture images of sachets retrieved using the Google Images search engine[2] and manually cropped to remove any background. Each scene is composed of many instances using the same texture image so as to make the occlusions between instances more challenging to detect. Besides, to prevent the network from simply substracting the background, we apply to the box a texture randomly chosen among 40 background images, retrieved using the Google Images search engine as well. A comprehensive overview of the textures and background images used for generating the Mikado dataset is provided in Figure 16. Between each image generation, we also randomly jitter the cameras and light locations to prevent the network from learning a fixed source of light, and so fixed reflections and shadows. The proposed dataset finally comprises on average 20.1 instances per image, hence 8 times more instances and 40 times more inter-instance occlusions per image than PIOD. Figure 3 provides samples and sums up the Mikado characteristics compared to the state-of-the-art datasets for oriented boundary detection (Fu et al, 2016; Wang and Yuille, 2016) and amodal instance segmentation (Follmann et al, 2019; Qi et al, 2019; Zhu et al, 2017).

Furthermore, to study the benefits of a richer synthetic data distribution, we make an extension of Mikado, namely Mikado+, following the same proposed generation pipeline but using more mesh templates (sachet, square sachet, box, cylinder-like shape), and more texture and background images. Figure 8a sums up the differences between Mikado and Mikado+.

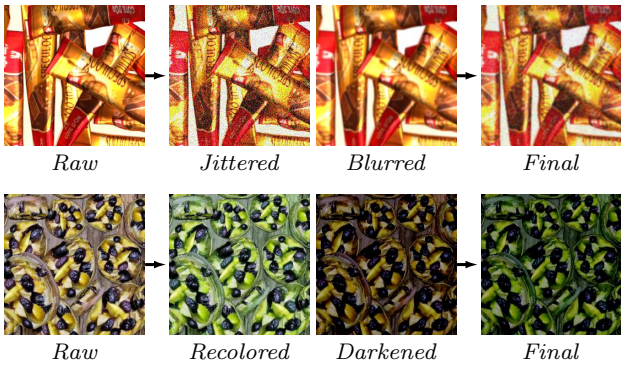
4.2 Data Augmentation

As our RGB images are generated using heuristic rendering models, the training and evaluation may be biased by a lack of realism in the sense that, unlike physical sensors and despite the variations of textures, deformations, and simulated specular reflections, a noise-free pixel information is provided to the network. To remedy this issue, we dynamically filter one image out of two with a gaussian blur and jitter independently the RGB values, as shown in Figure 8b, randomly at both training and testing times. The parameters for gaussian filtering and value jittering are randomly chosen within empirically

[2] https://images.google.com/

	Mikado	**Mikado+**
Mesh templates	1 	4    
Backgrounds	40	600
Textures	120	2,400
Images	2,400	14,560

(a) Offline augmentation.



(b) Online augmentation.

Fig. 8: Our synthetic data augmentation for Mikado and its extension Mikado+.

predefined intervals. This prevents the network from overfitting the too perfect synthetic color variations. In addition to dynamic blurring and RGB jittering, the Mikado+ images are also augmented with random permutation of the RGB channels and random under or over-exposition, as also illustrated in Figure 8b. Thus, Mikado+ depicts more color and lighting variations than Mikado.

We are aware of optimization-based data augmentation techniques out of the scope of this paper, such as the use of generative models (Antoniou et al, 2018) or automatic search to find the best augmentation policies (Cubuk et al, 2019). Nevertheless, our augmentation strategy is in line with the work of (Cubuk et al, 2019), for their search space consists of basic operations, such as rotation and color jittering, just as the ones that we manually apply on our synthetic images.

5 Experimental Setup

In this section, we describe our experiments to evaluate the proposed model and check the plausibility of the jointly proposed synthetic data. Specifically, the proposed model is evaluated on two differents aspects: (i) learning to map an image or a region that contains multiple overlapping similar instances to an instance-sensitive segmentation; (ii) learning to detect occlusion-aware instance boundaries. Our experiments are divided into three parts:

1. We compare variants of multicameral structures with alternative encoder-decoder designs, trained for occlusion-aware instance-sensitive segmentation.
2. We compare the bicameral part of our model with alternative layer and connection structurings, trained for occlusion-aware boundary detection.
3. We evaluate the plausibility of the proposed synthetic data on a real-world setup.

5.1 Evaluation Metrics

We use the same metrics to evaluate occlusion-aware segmentations and boundaries, as they all result from pixel-wise binary classification tasks. Specifically, we compute the precision and recall for different binarization thresholds, then typical derived metrics: the best F-score on dataset scale (ODS), the average precision (AP), and the average precision in high-recall regime (AP_{60}).

- ODS is the best harmonic mean of precision and recall over the full recall interval.
- AP conveys the area under the precision-recall curve over the full recall interval.
- AP_{60} is the average precision on the recall interval $[.6, 1]$, thus without taking into account high precisions due to empty inferences.

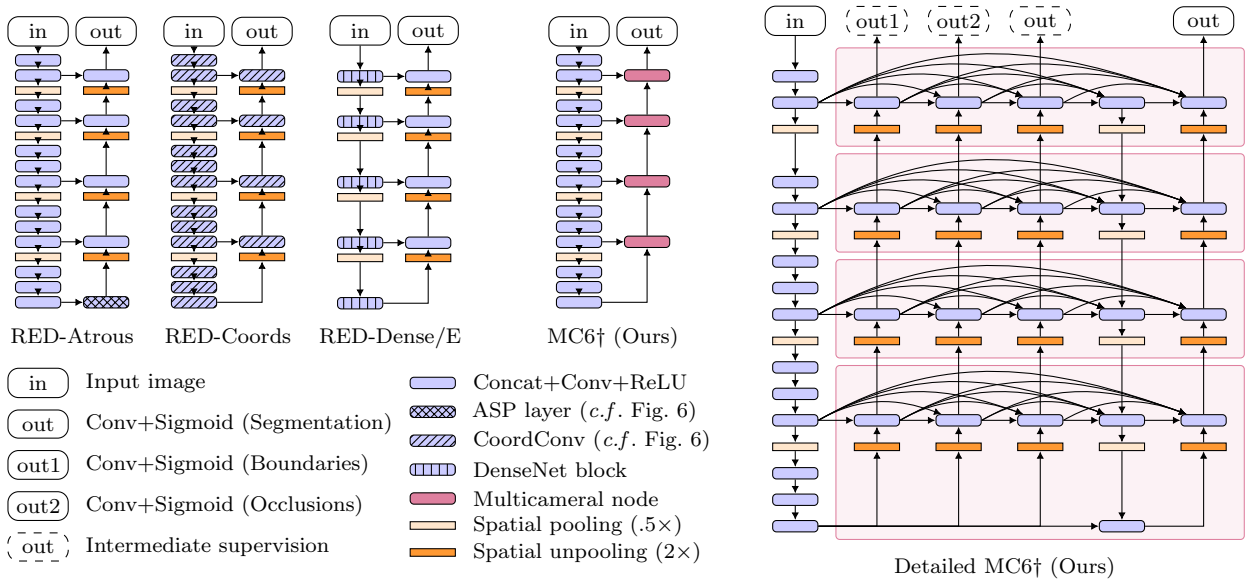
As matching tolerance, *i.e.* the maximum ℓ_2-distance to the closest ground-truth pixel for a positive or negative to be considered as true or false respectively, we set a hard value of 0 pixels for Mikado (which contains perfect ground-truth annotations) and a state-of-the-art value of $\tau = 0.0075\sqrt{W^2 + H^2} (\simeq 2.7$ pixels for 256×256 images) for PIOD and D2SA that contain approximative hand-made annotations, where $W \in \mathbb{N}^\star$ and $H \in \mathbb{N}^\star$ are the image width and height respectively. Evaluation is performed without non-maximum suppression, which may artificially improve precision.

5.2 Instance-Sensitive Segmentation

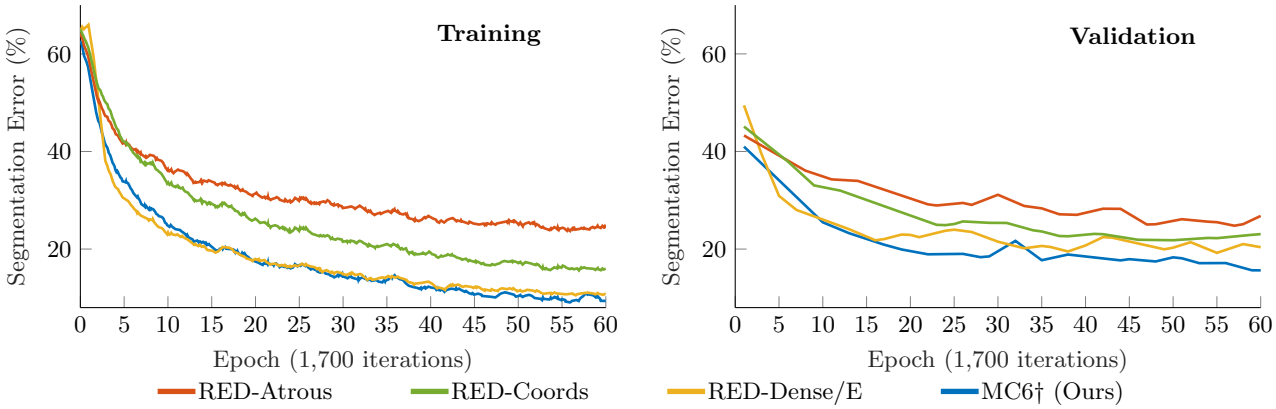
In our first set of experiments, we evaluate and analyze the proposed design for instance-sensitive segmentation on Mikado.

Baselines We first compare our design with state-of-the-art variants of residual encoder-decoder (RED) networks for reducing the translation invariance of the latent representations (see Figures 6 and 9).

- **Atrous spatial pyramid (Atrous)** Aggregating convolutions with different dilation rates on top of the encoder enables to capture longer-range pixel



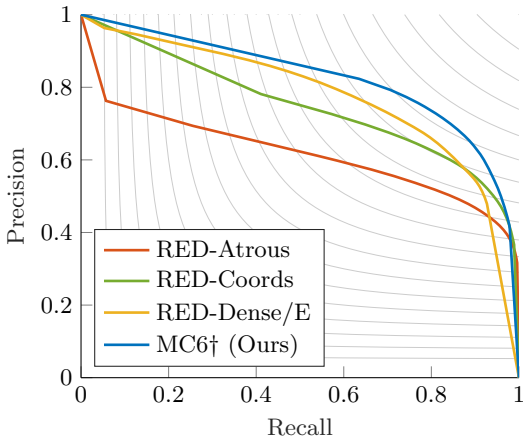
(a) Our multicameral structure compared with alternative design patterns for learning contextual representations.



(b) Comparative training and validation errors on Mikado.

Architecture	Number of parameters	Segmentation ODS	AP	AP_{60}
RED-Atrous	1,957,137	.631	.619	.506
RED-Coords	1,471,105	.703	.747	.599
RED-Dense/E	1,202,217	.724	.774	.593
MC6† (Ours)	5,411,916	**.767**	**.825**	**.691**
MC2(=RED)[4]	1,465,105	.696	.732	.587
MC3[4]	2,145,225	.705	.750	.598
MC4[4]	2,961,345	.709	.762	.609
MC4†[4]	2,961,747	.752	.802	.666
MC2-Coords/D[4]	1,490,113	.713	.754	.607
MC6†-Coords/D4[4]	5,417,916	**.766**	**.824**	**.696**
MC2-Atrous/D[4]	1,367,665	.591	.604	.454
MC4⋆†-Atrous/D2[4]	3,273,834	.609	.626	.476
MC6†-Atrous/D4[4]	5,053,356	**.784**	**.837**	**.706**

(c) Comparative performances on Mikado.



(d) Comparative precision-recall curves on Mikado.

[4] See Figure 10 for an overview of these architectures.

Fig. 9: Comparative results for occlusion-aware instance-sensitive segmentation on Mikado. In these experiments, a pruned VGG16 (or a pruned DenseNet121 for RED-Dense/E) is used as encoder backbone. Best viewed in color.

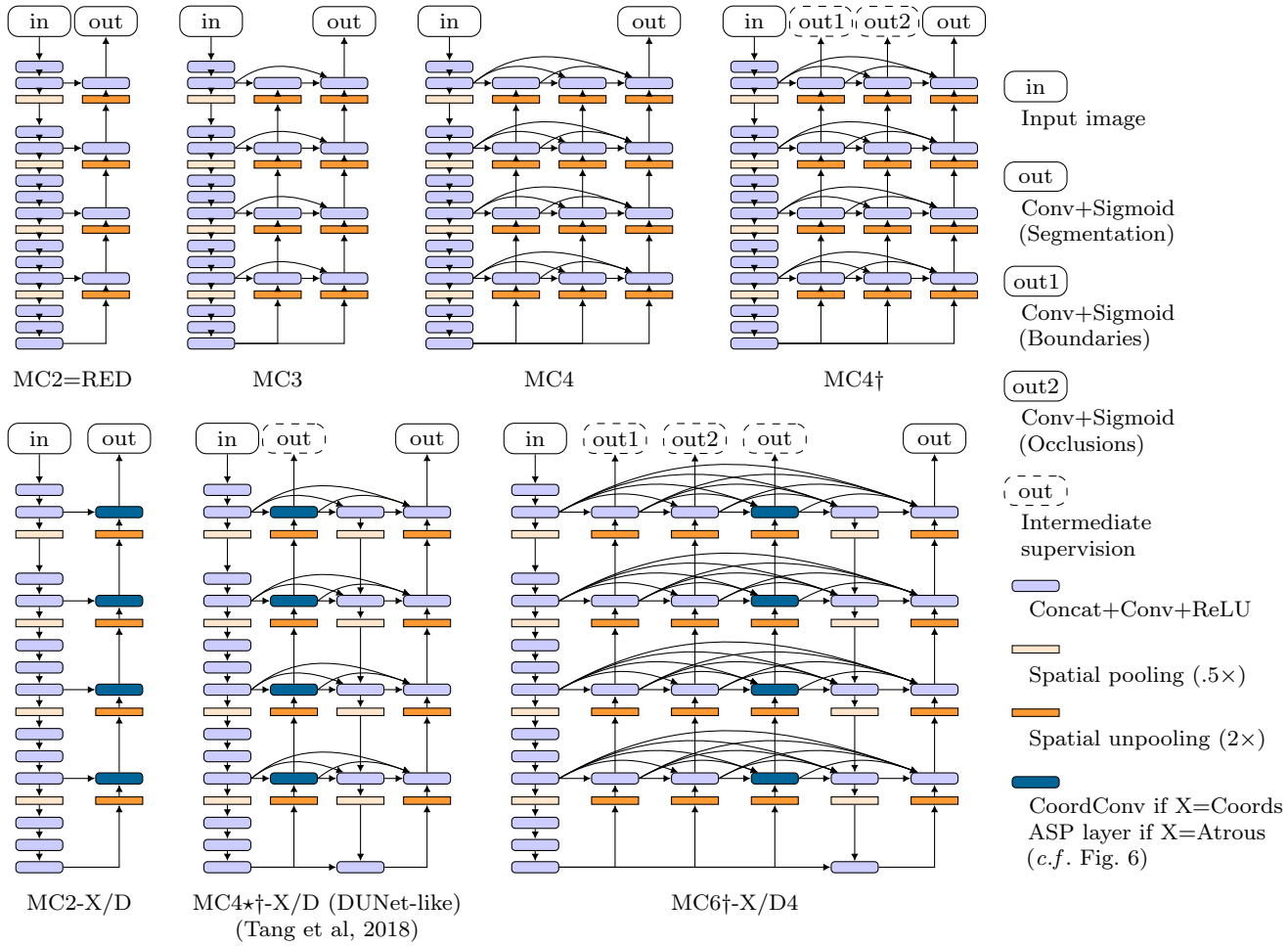


Fig. 10: Multicameral structures with different numbers of encoder and decoder units, and different node types for segmentation inference. Best viewed in color.

relations (Chen et al, 2018; Wang et al, 2018b; Yu and Koltun, 2016). Such relations are key cues to understand the notions of instance and occlusion. We compare with a RED network equipped with aggregated dilated convolutions on top of the encoder (RED-Atrous), similarly to (Chen et al, 2018).

– **Coordinate-aware convolutions (Coords)** Concatenating feature maps and hard-coded pixel coordinates, namely CoordConv, improves the learning of pixel classification tasks that require some translation variance (Liu et al, 2018b). We compare the proposed model with a RED network in which all the convolution layers are swapped to CoordConv ones (RED-Coords).
– **Dense encoder blocks (Dense/E)** Deepening the encoder blocks using densely connected layers has proved efficient for capturing more discriminative representations (Huang et al, 2017). Deeper hierarchical representations enable to encode more complex and longer-range pixel relations, as the receptive fields implicitly grow layer after layer. We include a RED network equipped with a DenseNet121-based encoder (RED-Dense/E) in our comparison.

Ablation study To further our evalution, we analyze three important aspects: the number of units in a multicameral sequence, the presence of intermediate supervisions, and the optional use of specific nodes in the decoding process. The resulting designs are illustrated in Figure 10.

– **Number of cascaded units** Adding decoder and encoder-decoder units in a multicameral sequence implies more parameters to train and more memory at inference time. We thus quantify the impact of many decoder units (MC2 vs. MC3 vs. MC4), and the presence of a refinement encoder-decoder unit (MC2 vs. MC4⋆†; MC4† vs. MC6†). Note that MC4⋆† is a periodic multicameral sequence of encoder-decoder units. This special case has been studied in (Tang et al, 2018), as DUNet, for refining visual landmark

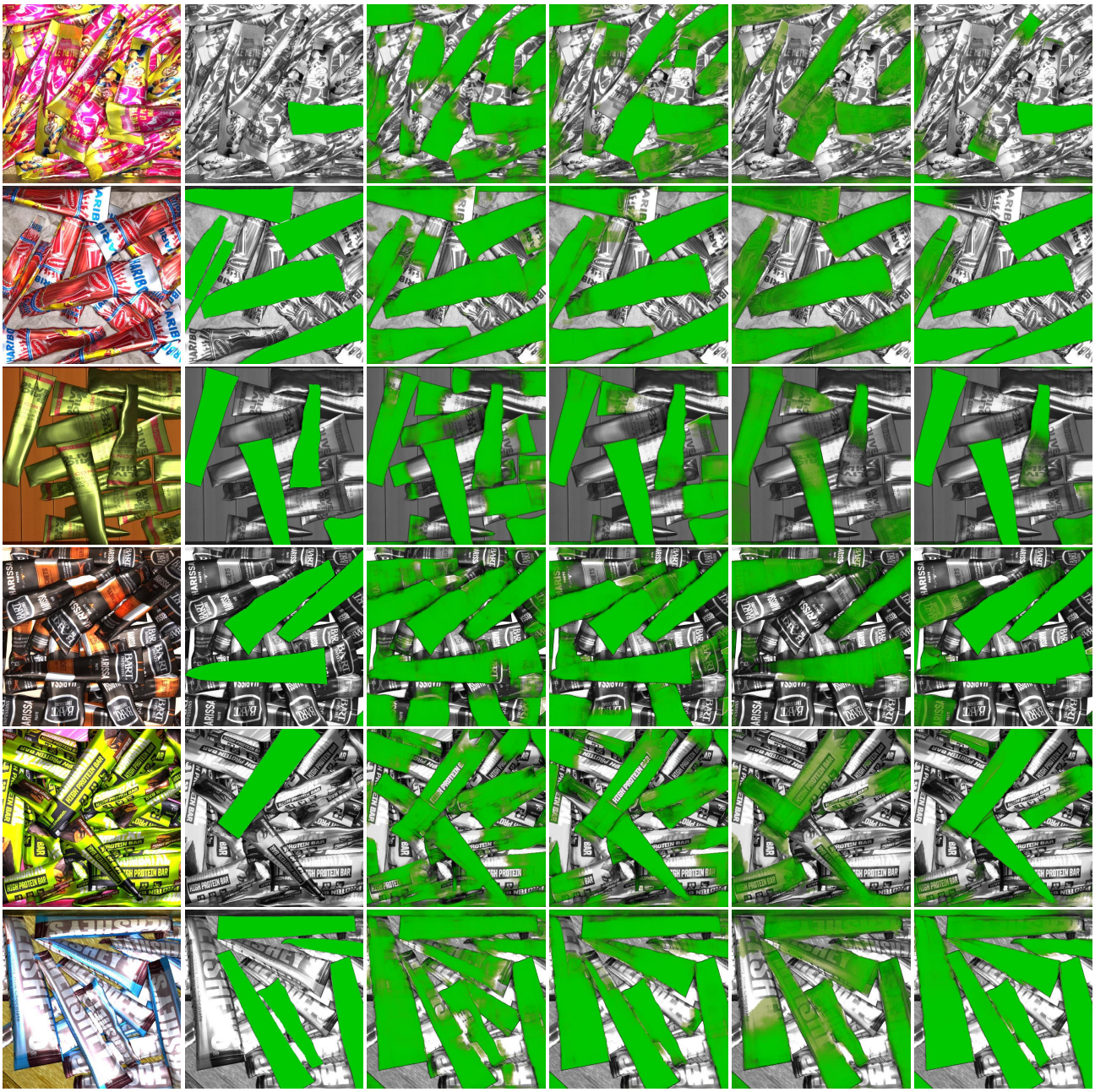
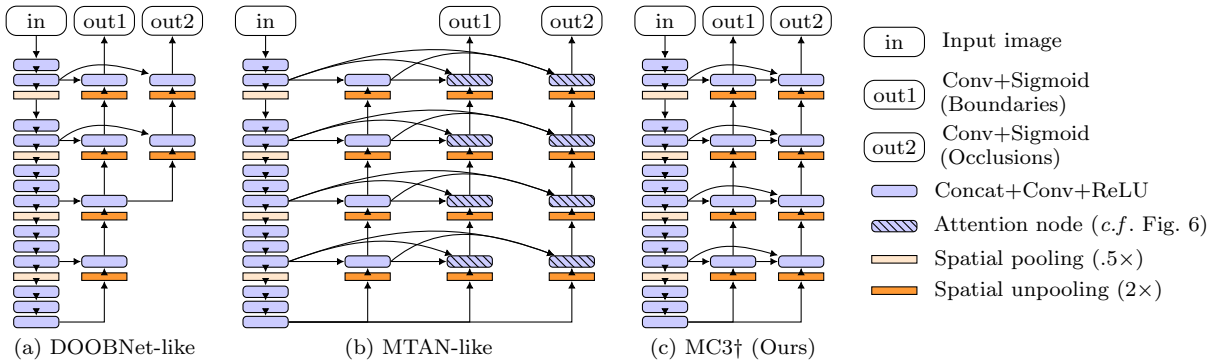


Fig. 11: Comparative results on Mikado using different encoder-decoder designs. Best viewed in color.

detection. Comparing MC4⋆† with MC6† therefore also shows the benefits of a more general coupling of units with ordinal intermediate supervisions.

– **Intermediate supervision** Generally, intermediate supervisions improve the training of complex graphs. In this work, we show the impact of ordinal intermediate supervisions to enforce a learning path: (1) detect image cues; (2) infer instance boundaries; (3) infer occluding boundary sides; (4) infer unoccluded instances. In our experiments, the first three decoders are supervised to infer the instance boundaries, the occluding boundary sides and the unoccluded instances respectively, using the loss functions presented in Section 3 (MC4† and MC6†). Comparing MC4 with MC4† thus shows the impact of such supervisions.
– **Optional specific nodes** Dilated and coordinate-aware convolutions locally reduce the translation



(a) DOOBNet-like (b) MTAN-like (c) MC3† (Ours)

Dataset:		Mikado						PIOD					
Architecture	**Number of parameters**	**Boundaries**			**Occlusions**			**Boundaries**			**Occlusions**		
		ODS	AP	AP_{60}	ODS	AP	AP_{60}	ODS	AP	AP_{60}	ODS	AP	AP_{60}
DOOBNet-like	1,497,330	.703	.764	.583	.729	.806	.633	.639	.674	.446	.629	.669	.451
MTAN-like	2,075,546	.700	.762	.579	.727	.809	.628	.646	.683	.437	.632	.676	.444
MC3† (Ours)	2,145,426	.701	.762	.581	**.737**	**.815**	**.645**	.642	.673	.450	**.633**	**.683**	**.454**

Comparative performances on Mikado and PIOD.

Fig. 12: A bicameral structure (MC3†) compared with state-of-the-art design patterns adapted for occlusion-aware boundary detection. (a) Encoder and low-resolution half-decoder shared by two independent high-resolution half-decoders. (b) Task-specific decoders with attention mechanisms to select shared features. (c) Encoder shared by two cascaded decoders. In these experiments, a pruned VGG16 is used as encoder backbone. Best viewed in color.

invariance of convolutional embeddings. We try to combine these design patterns within our multicameral sequence. Specifically, we compare variants of MC2 and MC6† networks in which we use such nodes in the first decoder for outlining the unoccluded instances (D and D4 respectively). These variants are thus referred to as MC2-X/D and MC6†-X/D4 respectively, with X ∈ {Coords,Atrous}.

Implementation details Due to hardware limitations, we compare the networks using a pruned VGG16 (or a pruned DenseNet121 for the RED-Dense/E design) as first encoder backbone. Specifically, we keep the first quarter of filters at each layer in the original encoder. For the remaining layers, we set a kernel size of 5×5 and the numbers of filters reported in Table 1.

		E_s^1		$\{E, D\}_s^{t>1}$	
Resolution	VGG16	full	pruned	full	pruned
$s = 1$	conv1_x	64	16	32	8
$s = 2$	conv2_x	128	32	64	16
$s = 3$	conv3_x	256	64	128	32
$s = 4$	conv4_x	512	128	256	64
$s = 5$	conv5_x	512	128	256	64

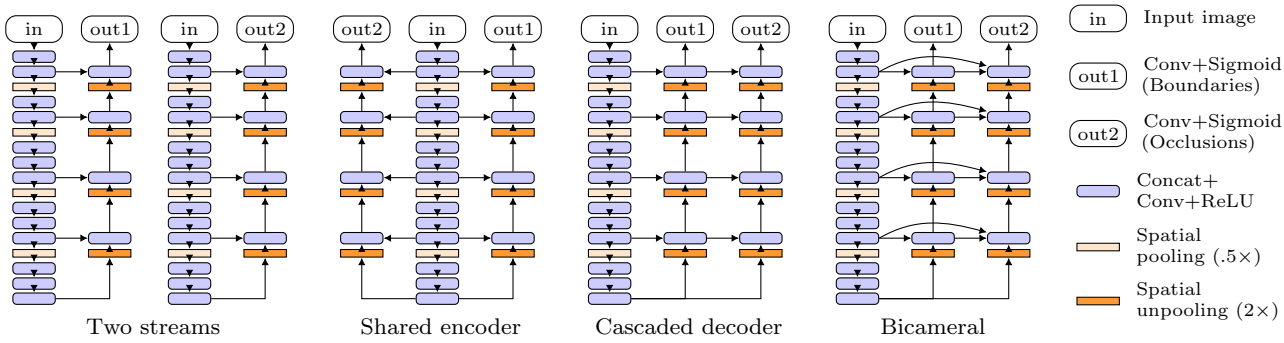
Table 1: Number of filters for each layer in our full or pruned network implementations, using a full or pruned VGG16 as first encoder backbone (E_s^1).

5.3 Occlusion-Aware Boundaries

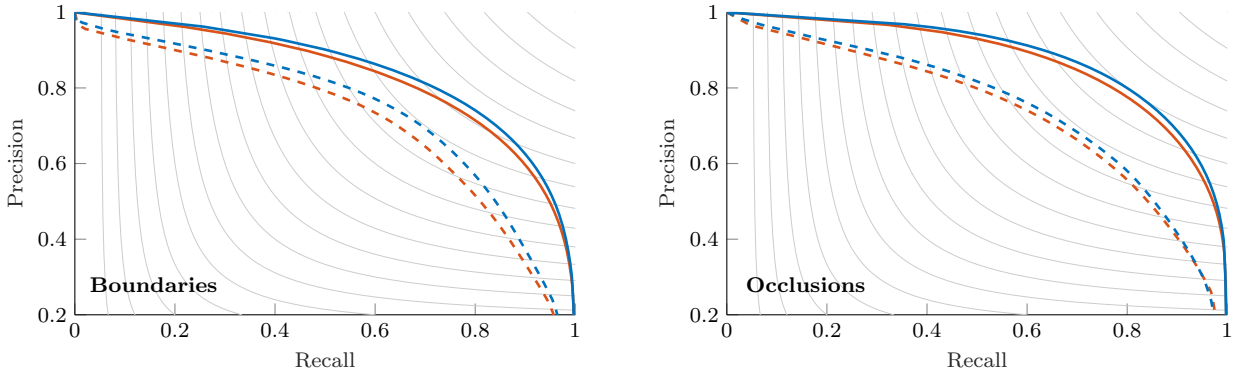
Our most performance-enhancing multicameral design (MC6†) includes a bicameral structure (MC3†) trained for occlusion-aware boundary detection. To further our analysis on the multicameral components, we evaluate this structure alone on Mikado and PIOD.

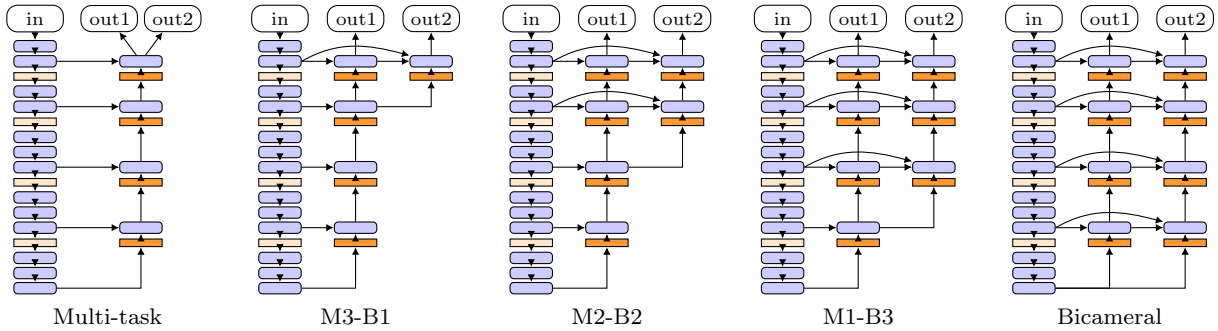
Baselines We compare MC3† with related layer and connection structurings, released concurrently to our work (see Figure 12).

- **DOOBNet** (Wang et al, 2018a) proposed an incremental improvement of (Wang and Yuille, 2016) for occlusion-aware boundary detection. (Wang and Yuille, 2016) employed two independent VGG16-based encoder-decoder networks for boundaries and occlusion orientations respectively. Instead, (Wang et al, 2018a) used a single encoder and a single low-resolution half-decoder, both shared by two independent high-resolution decoders. They also proposed incremental improvements: a ResNet-based encoder, an ASP layer on top of it like in (Chen et al, 2018), and a focal loss-like function to drive the training (Lin et al, 2017). We compare a bicameral structure with the core DOOBNet design, *i.e.* without these incremental improvements.
- **MTAN** In a more general context, (Liu et al, 2019) have introduced attention masks at each resolution for pixel-wise multi-task learning. Such masks enable



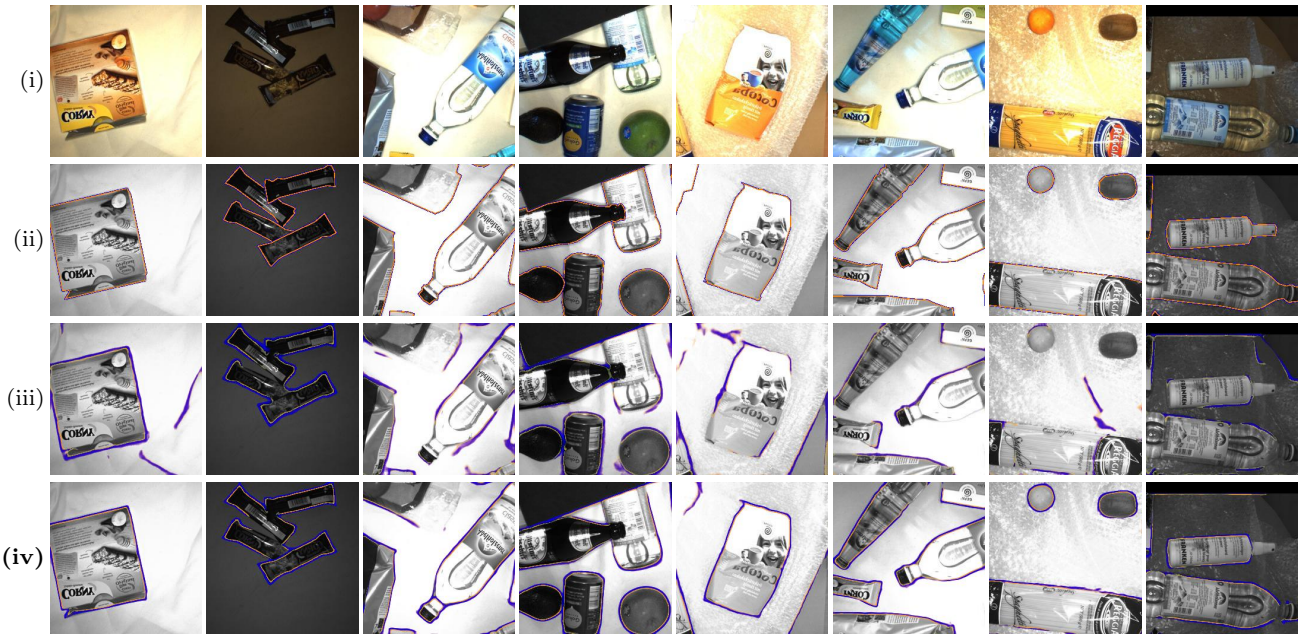
Dataset:		Mikado				PIOD			
Architecture	**Number of parameters**	**Boundaries**		**Occlusions**		**Boundaries**		**Occlusions**	
		ODS	AP	ODS	AP	ODS	AP	ODS	AP
Two streams	46,839,938 (×1.0)	.755	.832	.788	.872	.673	.708	.681	.733
Shared encoder	32,125,250 (×.69)	.769	.847	.792	.876	.692	.732	.686	.738
Cascaded decoders	29,949,250 (×.64)	.766	.844	.795	.880	.694	.735	.689	.748
Multi-task decoder	23,420,770 (×.50)	.767	.845	.795	.880	.691	.731	.679	.731
Bicameral (=MC3†)	34,301,250 (×.73)	**.769**	**.847**	**.801**	**.884**	**.697**	**.738**	**.692**	**.747**



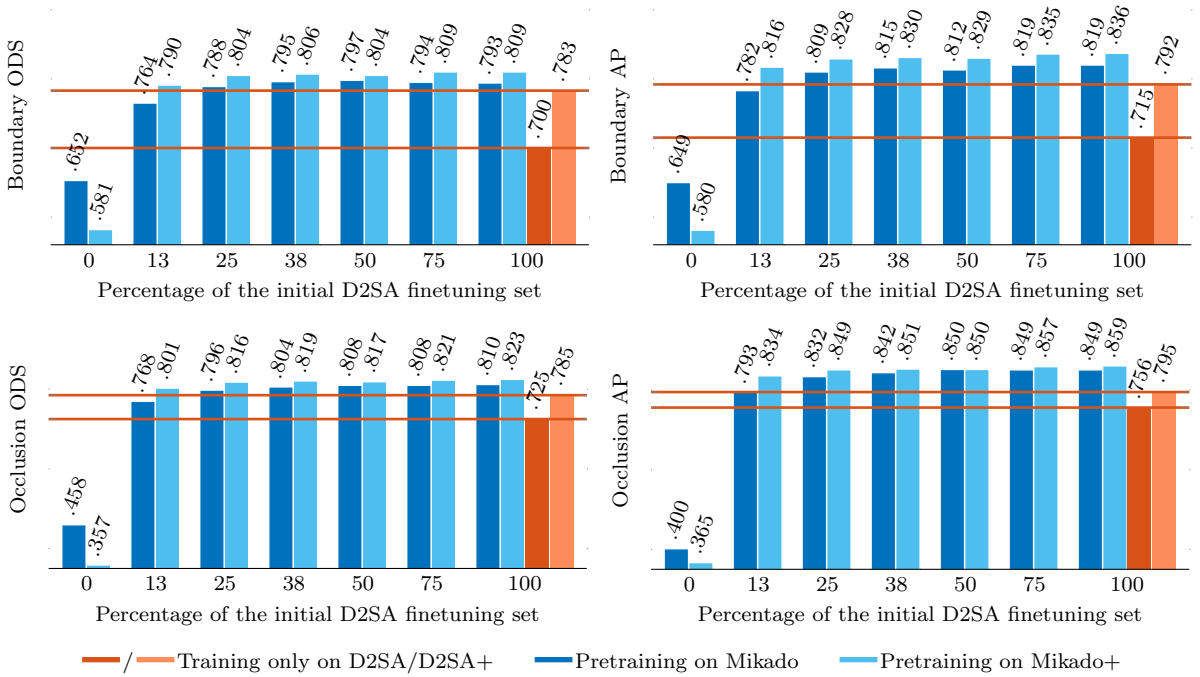


Dataset:		Mikado				PIOD			
Architecture	**Number of parameters**	**Boundaries**		**Occlusions**		**Boundaries**		**Occlusions**	
		ODS	AP	ODS	AP	ODS	AP	ODS	AP
Multi-task	23,420,770 (×.50)	.767	.845	.795	.880	.691	.731	.679	.731
M3-B1 hydrid	23,548,802 (×.50)	.767	.845	.796	.879	.691	.735	.683	.734
M2-B2 hydrid	24,060,866 (×.51)	.769	.848	.797	.881	.692	.738	.685	.740
M1-B3 hydrid	26,108,994 (×.56)	**.771**	**.848**	**.802**	**.885**	.693	.737	.685	.739
Bicameral (=MC3†)	34,301,250 (×.73)	.769	.847	.801	.884	**.697**	**.738**	**.692**	**.747**

Fig. 13: Ablation study on a bicameral structure for occlusion-aware boundary detection. In these experiments, a full VGG16 is used as encoder backbone. The best overall performances are obtained by sharing a single encoder and cascaded decoders, altogether linked via resolution-wise skip connections. Best viewed in color.



(a) Comparative results for instance boundary (blue) and occluding boundary side (orange) detection on D2SA. From top to bottom: input (i), ground truth (ii), prediction using the proposed network trained on D2SA (iii), using the proposed network pretrained on Mikado then finetuned on D2SA with the first three encoder blocks frozen (iv). Pretraining the proposed network on Mikado before finetuning on D2SA leads to significant improvements.



(b) Performances of a bicameral network pretrained on Mikado/Mikado+ then finetuned on D2SA with the encoder blocks 1, 2, 3 frozen (see also Figure 20 in appendix). The performances are shown w.r.t. the percentage of real images retained for finetuning. Exploring a wider range of configurations in simulation (Mikado+) enables to learn more abstract local representations of the boundaries and occlusions, thus achieving state-of-the-art performances while drastically reducing the number of real images for finetuning.

Fig. 14: Comparative results on D2SA using a bicameral structure trained for occlusion-aware boundary detection, under different pretraining conditions. Best viewed in color.

resolution-wise task-specific selections of shared features. As learning jointly boundaries and occlusions also requires shared and task-specific representations, we compare bicameral decoders with MTAN-like decoders for boundaries and occlusions respectively.

Ablation study To further our above comparison, we isolate the impacts of sharing a single encoder and cascading decoders, and we study how bicameral decoders compare with partially shared decoders (*c.f.* Figure 13). In appendix, we also study the impact of bicameral skip connections (see Figures 18 and 19).

- **Bicameral components** We compare a bicameral structure with three intermediate designs: two independent encoder-decoder streams (DOC-like (Wang and Yuille, 2016)); two independent decoders sharing a single encoder; two cascaded decoders sharing a single encoder.
- **Partial decoder sharing** We compare a bicameral structure with four alternative levels of decoder sharing: bicameral decoders sharing their lowest-resolution layer; sharing their two lowest-resolution layers; their three lowest-resolution ones; all their layers, which is equivalent to multi-task decoding.

Implementation details We use a pruned VGG16 as encoder backbone for our comparison with DOOBNet-like and MTAN-like architectures. In our ablation study, a full VGG16 is used as encoder backbone. Our pruning scheme and layer hyperparameters are the same as the ones in Section 5.2.

5.4 Data Plausibility Check

As Mikado is a computer-generated dataset, one may raise the question whether it is realistic. The answer is obviously no, but we claim that it is valuable for significative evaluations. To prove this point, we evaluate the transferability of features learned from Mikado to real data. In line with (Yosinski et al, 2014), features learned from a source domain are transferable if they can be repurposed and boost generalization on a target domain. As target domain, we use D2SA (Follmann et al, 2018) (see samples in Figure 3).

Synthetic feature transferability As deep features transition from general to specific by the last layers, we train a bicameral network for occlusion-aware boundary detection on Mikado, then freeze some of the encoder blocks and retrain the remaining layers on D2SA. We conduct different finetunings, by reducing progressively the number of D2SA images used for finetuning.

Synthetic data distribution To highlight the benefits of synthetic data in contrast with hardly extensible real-world datasets, we additionally study how a richer synthetic data distribution, *i.e.* Mikado+, impacts the domain adaptation. As the ranges of texture, shape, and pose variations are more widely represented in Mikado+, better transferable invariants are expected to be learned. In a limited manner, D2SA addresses this case by overlaying manually isolated instances into fake training images (Follmann et al, 2018). We thus compare with this augmentation strategy, referred to as D2SA+.

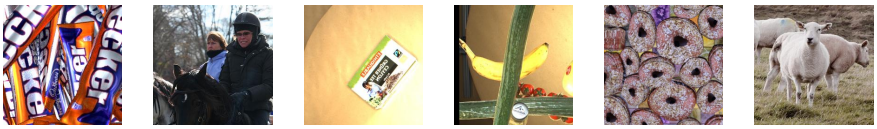
Implementation details To expose the most transferable features learned from Mikado, we first compare bicameral networks finetuned on D2SA with different encoder block at which the network is chopped and retrained (*c.f.* Fig. 20 in appendix). We define a block as a set of convolutional layers between two pooling layers. A VGG16-based encoder is therefore composed of 5 blocks. A block is said "frozen" when the corresponding parameters remain unchanged during finetuning. Note that the choice of the layers to freeze is application-dependent because the levels of semantics to freeze depend on the differences between the source and target domains.

Note also that we consider D2SA instead of PIOD or COCOA for transfer learning from Mikado because the data distributions of PIOD and COCOA are very different from Mikado. Indeed, (Ben-David et al, 2010a,b) show that a low divergence between the source and target domain distributions is a necessary condition for the success of domain adaptation. Table 17c in appendix empirically shows that this condition is not met for Mikado and PIOD. Unlike PIOD and COCOA, which contain natural images of indoor and urban scenes with people, cars and animals, D2SA and Mikado both contain top-view images of household objects in bulk.

5.5 Training Settings

Each network is trained and tested in the same conditions (including fixed random seeds) using Caffe (Jia et al, 2014).

Data preparation The networks are not fed with the original images but 256×256 sub-images randomly extracted from each original image, and augmented offline with random geometric transformations (flipping, scaling and rotation). The folds of Mikado and Mikado+ are defined such that a texture appears only in one of the three subsets. The folds of PIOD and D2SA are defined with respect to the initial split proposed by their authors. Specifically, the original training images are used for training or validation in our folds, and the original



Dataset:	**Mikado**	**PIOD**	**D2SA**	**D2SA+**	**Mikado+**	**COCOA**
Training images	13,600	9,600	512	2,960	28,800	12,800
Validation images	800	800	56	328	4,800	1,424
Test images	4,800	800	5,992	5,992	–	1,323
Iterations per epoch	1,700	1,200	64	370	3,600	1,600

Table 2: Image folds for each dataset after offline augmentation.

validation images for test. The original test images are never used as they are not publicly available.

Optimization We use the Adam solver (Kingma and Ba, 2015) with $\beta_1 = .9$, $\beta_2 = .999$, $\epsilon = 10^{-8}$, and an initial learning rate of 10^{-4}. We add a ℓ_2-regularization with a weight decay of 10^{-4}. The batch size is set to 8, and the training images are randomly permuted at each epoch. Since we solve a non-convex optimization problem, without theoretical convergence guarantees, the number of training iterations is chosen for each dataset from an empiric analysis on training and validation subsets. As generally adopted, the optimization is stopped when the validation error stagnates or increases while the training error keeps decreasing.

- In our comparative experiments (Figures 9 and 12), we stop each training after 60 epochs for both Mikado and PIOD. Due to hardware limitations, each score results from one data fold.
- In our ablation study on bicameral structuring (Figure 13), each optimization is stopped after 20 and 15 epochs for Mikado and PIOD respectively, and each score is averaged over three optimizations using different data folds.
- In our transfer learning experiments (Figure 14), each finetuning on D2SA is stopped after 15 epochs, and each score is averaged over three optimizations using different data folds. Pretraining on Mikado+ is stopped after 30 epochs.

Details on the epochs and data folds for each dataset are provided in Table 2. Please note that although the chosen stopping criterion may not be optimal for reaching the best performances on each dataset, it is however sufficient for significative comparisons since each network is trained under the same conditions.

Initialization For all experiments, except finetuning from weights pretrained on Mikado or Mikado+ in our synthetic data plausiblity check, each network has its first encoder initialized with weights pretrained on ImageNet (Russakovsky et al, 2015), and the remaining layers with the Xavier method (Glorot and Bengio, 2010). To avoid overfitting, each convolutional block is ended with a dropout layer (we set the dropout ratio to .5), except in the first encoder.

6 Discussion

In this section, we argue in light of our experimental results that the proposed multicameral decoder is more effective for dense homogeneous layouts than alternative design patterns, and that the jointly proposed synthetic data is plausible with respect to real-world problems.

6.1 On the Proposed Model

Homogeneous layouts require a complex decoding process. When localizaling specific instances in dense homogeneous layouts, the decoding process has great importance because the pixel embeddings must discriminate between instances of the same object. Figure 9 confirms that a multicameral design proves more effective on Mikado than state-of-the-art design patterns for capturing position-sensitive representations. Specifically, our MC6† design outperforms RED-Atrous, RED-Coords, and RED-Dense/E networks by 20.6, 7.8 and 5.1 points in AP respectively. We explain these differences as follows: RED-Atrous enlarges the receptive field at the lowest resolution, which may lead to overfitting the training object layouts or mistakenly capturing relations between similar patterns far away from each other; RED-Coords associates each latent representation with a global location, thereby reducing the generalizability of these representations; RED-Dense/E uses DenseNet121 encoder blocks to softly capture more complex image representations that can hardly be fully exploited within a simple decoding process. Using only a VGG16 encoder, our multicameral decoding process produces higher-quality segmentations and more contrasted pixel-wise decisions, as illustrated in Figure 11. Nevertheless, half-outlined instances still appear (see the third row of Figure 11),

seemingly due to a lack of long-range pixel associations in the learned representations.

Structured decoding units improves the learning. The success of a multicameral design results from our design choices to structure the decoding process: cascading subtask-specific decoder and encoder-decoder units. As reported by Figure 9c, cascading simple decoders without intermediate supervisions gradually improves the performances. Starting from MC2, adding one decoder (MC3) increases AP by 1.8 points, adding another decoder (MC4) by 3 points. Furthermore, structuring the backpropagation signals with ordinal intermediate supervisions for instance boundary and occluding boundary side detections (MC4†) enables an additional gain of 4 points. Finally appending an encoder-decoder unit for refining the segmentation (MC6†) leads to an overall pixel-wise improvement of 9.3 points over MC2, a VGG16-based RED network without additional state-of-the-art components. All these experimental results confirm that encouraging subtask-specific feature through ordinal multiscale units is an effective design pattern for dense homogeneous layouts.

Learning position-sensitive representations proves more effective late in the decoding process. A multicameral design can be enhanced by enlarging the receptive fields just before decoding the unoccluded instances (MC6†-Atrous/D4). As reported by Figure 9c, MC6†-Atrous/D4 outperforms MC6† by 1.2 points. Learning explicity position-sensitive representations late in the decoding process enhances the performances in alternative design upgrades. Specifically, Figure 9c reports various similar improvements. First, using coordinate-aware convolutions: between RED-Coords and MC2-Coords/D (note that MC2 and RED are equal); between MC2-Coords/D and MC6†-Coords/D4. Second, using dilated convolutions: between MC2-Atrous/D and MC4⋆†-Atrous/D2; between MC4⋆†-Atrous/D2 and MC6†-Atrous/D4. These observations strongly suggest that the use of position-sensitive transforms, which partially break the translation invariance property of convolutional layers, should be thought with respect to the convolutional and non-convolutional aspects of the learned task. We applied this principle in our MC6†-Atrous/D4 design: instance-aware segmentation requires some translation variance, while occlusion-aware boundary detection does not.

Ordinal decoders are important for detecting occlusion-aware boundaries as well. Our discussion on the importance of structure decoding extends to the lower-level task of occlusion-aware instance boundary detection. As reported by Figure 12, a bicameral network trained for jointly detecting instance boundaries and occluding boundary sides (MC3†) compares favorably with DOOBNet-like and MTAN-like designs. Specifically, our design increases AP in the high-recall regime for occlusions by 1.7 points and 1 point on Mikado and PIOD respectively. Indeed, a key difference between MC3† and these state-of-the-art structurings is the ordinal relation between our decoders to encourage subtask-specific feature reuse. A bicameral structure is particularly suited to occlusion-aware boundary detection because occluding boundary sides can be interpreted as instance boundaries translated in the direction of the occluding instance.

Our ablation study on bicameral structuring (Figure 13) confirms this important aspect. Specifically, a bicameral structure, which combines a shared encoder and cascaded decoders, achieves the best overall performances on both Mikado and PIOD. A bicameral structure also compares favorably with bicameral decoders that partially share their layers.

6.2 On the Proposed Synthetic Data

Mikado enables a meaningful evaluation. We create Mikado for our evaluation because, to the best of our knowledge, dense homogeneous layouts are missing from the public datasets for occlusion-aware instance segmentation. Although Mikado is a synthetic dataset, it is valuable for a meaningful evaluation. Our experimental results in Figure 14 show that Mikado enables transferable feature learning in line with (Yosinski et al, 2014). Specifically, we show that using synthetic representations learned from Mikado enables to better detect occlusion-aware instance boundaries on D2SA (Follmann et al, 2018). As reported by Figure 14b, a gain of more than 10 points in AP for boundaries and 9 points for occlusions is achieved when finetuning the proposed network on D2SA with the first three encoder blocks frozen after pretraining on Mikado, instead of training all the layers only on D2SA (see also Figure 20 in appendix). This gain is qualitatively corroborated by Figure 14a. It suggests that a network trained on Mikado, which contains more occlusion relations between instances than the D2SA images for finetuning, learns a more general notion of occlusion. Our simulation-based pretraining also proves more effective than D2SA+ (Follmann et al, 2018), *i.e.* creating training images by overlaying manually isolated instances. Despite the domain shift between Mikado and D2SA, using simulation enables more physics-consistent rendering at boundaries and less redundancy in terms of poses, unlike brute-force overlaying of instance segments from real images. Furthermore, almost equivalent performances are achieved when reducing the number of human-labeled real images for

finetuning. Figure 14b shows that a bicameral network finetuned on D2SA using only 25% of the initial D2SA finetuning subset, with the first three encoder blocks frozen after pretraining on Mikado, still outperforms a bicameral network trained only on D2SA or D2SA+. All of these results confirm that the representations learned from Mikado are meaningful w.r.t. real-world setups.

Mikado+ leads to even better results. Unlike real-world datasets, a synthetic dataset is readily extensible. By enriching Mikado with 20 times more texture images, 15 times more background images and 4 mesh templates, namely Mikado+, the ranges of color, texture, shape, and pose variations are better represented. As shown by Figure 14b, this leads to more generalizable invariants. Specifically, pretraining on Mikado+ instead of training only on D2SA increases AP by 10.1 points for boundaries and 7.8 points for occlusions while using only 12.5% of the initial D2SA finetuning set. By contrast, using Mikado in the same conditions leads to a gain of 3.4 points for boundaries and 4.1 points for occlusions. These results imply that Mikado+ enables to learn more abstract local representations than Mikado. However, when applied on D2SA without finetuning, a pretraining on Mikado+ proves less effective than on Mikado. Consistently with the results after finetuning on D2SA, this could be explained by an overgeneralization of the task-specific layers. The neurons indeed co-adapt to capture the most discriminative patterns that are not likely to be the colors nor the object and background textures in Mikado+. An over-randomization of the colors and textures may disconnect the learned representations from concrete examples. This has nevertheless the advantage of easing the finetuning on D2SA, as the real-world scenes then appear as one variation within the learned range of variations. All these observations are incentives to favor synthetic training data when pixel-wise annotations on real-world images are hardly collectable. Hand-made annotations may also hinder the training due to their inaccuracy and incompleteness. As illustrated by Figure 17b in appendix, a bicameral network trained on PIOD is able to fairly predict non-annotated boundaries, *e.g.* internal boundaries of instances with holes, missing instances, or instances ambiguously considered as part of the background. Furthermore, objects with complex shape, such as houseplants, which are often coarsely annotated by humans, are finely delineated by the proposed network.

7 Conclusion

We aimed at outlining unoccluded instances in dense homogeneous layouts, using a deep residual encoder-decoder design. However, decoding translation-invariant representations becomes problematic for distinguishing identical instances. Unlike the state-of-the-art solutions which strengthen the encoder while reducing the decoder to a mere upsampling branch, we increased the complexity in the decoder by coupling decoder and encoder-decoder units in cascade, using resolution-wise skip connections. We also introduced a synthetic data generation pipeline (Mikado) to produce images of dense homogeneous layouts, as this scenario is missing from the public datasets. Our experiments on Mikado and PIOD showed that: (i) a multicameral design gives better results than aggregated dilated or coordinate-aware convolutions; (ii) ordinal multiscale latent representations improve the attention to unoccluded instances; (iii) design patterns for reducing the translation invariance are more efficient later in the decoding process. Furthermore, our experiments on transfer learning from Mikado to D2SA showed that a pretraining on Mikado enables state-of-the-art performances, while reducing by more than 85% the number of real images for finetuning.

The proposed synthetically pretrained multicameral FCN establishes a new baseline for parsing images of dense homogeneous layouts. Nevertheless, there are still open research directions. Due to the "horizontal" skip connections, the number of filters severely increases with the number of decoding units, which may be prohibitive in terms of computational cost and memory requirements. It would be worth investigating optimization-based strategies, such as network architecture search approaches (Cai et al, 2019; Yu et al, 2019), to determine the optimal grid node and subtask ordering with respect to the application. Executing the model on the image at a lower-resolution then using adaptive sparse representations to iteratively refine the inferred boundaries could be another path to explore, as suggested by (Kirillov et al, 2019). Furthermore, the proposed model does not explicitly exploit the redundancy within the scene. Yet, instances of the same object provide many cues to build an implicit object representation. Explicitly capturing the correspondences between the instances of a pile could be achieved using graph convolutional modules, in the same vein as dual graph networks for heterogeneous scenes (Zhang et al, 2019). Finally, a pretraining on Mikado requires some domain adaptation to achieve expert-level performances on a specific application. Although the proposed pretraining drastically reduces the need of annotations, producing the segmentation of a dense layout manually is very tedious. Coupling the proposed learning with a generative adversarial network (Dong et al, 2018) or using self-supervision (Lee et al, 2019) would enable ordinal decoder units to adapt to novel conditions from unlabeled images.

Acknowledgements We thank Romain Brégier, Florian Sella and the anonymous reviewers for their insightful comments and suggestions that helped us to greatly improve this article.

Note: This is a pre-print of an article published in International Journal of Computer Vision, Special Issue on Deep Learning for Robotic Vision. The final authenticated version is available online at:
https://doi.org/10.1007/s11263-020-01323-0

References

Antoniou A, Storkey AJ, Edwards H (2018) Augmenting Image Classifiers Using Data Augmentation Generative Adversarial Networks. *In: International Conference on Artificial Neural Networks and Machine Learning (ICANN)*, Springer, Lecture Notes in Computer Science, vol 11141, pp 594–603

Ayvaci A, Raptis M, Soatto S (2010) Occlusion Detection and Motion Estimation with Convex Optimization. *In: Advances in Neural Information Processing Systems (NIPS)*, pp 100–108

Ayvaci A, Raptis M, Soatto S (2012) Sparse Occlusion Detection with Optical Flow. *International Journal of Computer Vision (IJCV)* 97(3):322–338

Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39(12):2481–2495

Bai M, Urtasun R (2017) Deep Watershed Transform for Instance Segmentation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 2858–2866

Batra A, Singh S, Pang G, Basu S, Jawahar C, Paluri M (2019) Improved Road Connectivity by Joint Learning of Orientation and Segmentation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, Computer Vision Foundation / IEEE, pp 10385–10393

Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010a) A theory of learning from different domains. *Machine Learning* 79(1-2):151–175

Ben-David S, Lu T, Luu T, Pál D (2010b) Impossibility Theorems for Domain Adaptation. *In: International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR.org, JMLR Proceedings, vol 9, pp 129–136

Blender Online Community (2016) Blender - a 3D modelling and rendering package. Blender Foundation, Blender Institute, Amsterdam, URL http://www.blender.org

Brégier R, Devernay F, Leyrit L, Crowley JL (2017) Symmetry Aware Evaluation of 3D Object Detection and Pose Estimation in Scenes of Many Parts in Bulk. *In: International Conference on Computer Vision Workshops (ICCVW)*, IEEE Computer Society, pp 2209–2218

Caesar H, Uijlings JRR, Ferrari V (2018) COCO-Stuff: Thing and Stuff Classes in Context. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 1209–1218

Cai H, Zhu L, Han S (2019) ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. *In: International Conference on Learning Representations (ICLR)*

Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *In: European Conference on Computer Vision (ECCV) Part VII*, Springer, Lecture Notes in Computer Science, vol 11211, pp 833–851

Cubuk ED, Zoph B, Mane D, Vasudevan V, Le QV (2019) AutoAugment: Learning Augmentation Strategies From Data. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, Computer Vision Foundation / IEEE, pp 113–123

Dai J, He K, Sun J (2016) Instance-Aware Semantic Segmentation via Multi-task Network Cascades. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 3150–3158

Deng R, Shen C, Liu S, Wang H, Liu X (2018) Learning to Predict Crisp Boundaries. *In: European Conference on Computer Vision (ECCV) Part VI*, Springer, Lecture Notes in Computer Science, vol 11210, pp 570–586

Do TT, Nguyen A, Reid ID (2018) AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection. *In: International Conference on Robotics and Automation (ICRA)*, IEEE, pp 1–5

Dong X, Yan Y, Ouyang W, Yang Y (2018) Style Aggregated Network for Facial Landmark Detection. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 379–388

Eigen D, Puhrsch C, Fergus R (2014) Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. *In: Advances in Neural Information Processing Systems (NIPS)*, pp 2366–2374

Everingham M, Eslami SM, Gool L, Williams CK, Winn J, Zisserman A (2015) The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)* 111(1):98–136

Fan R, Cheng MM, Hou Q, Mu TJ, Wang J, Hu SM (2019) S4Net: Single Stage Salient-Instance Segmentation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, Computer Vision Foundation / IEEE, pp 6103–6112

Follmann P, Böttger T, Härtinger P, König R, Ulrich M (2018) MVTec D2S: Densely Segmented Supermarket Dataset. *In: European Conference on Computer Vision (ECCV) Part X*, Springer, Lecture Notes in Computer Science, vol 11214, pp 581–597

Follmann P, König R, Härtinger P, Klostermann M, Böttger T (2019) Learning to See the Invisible: End-to-End Trainable Amodal Instance Segmentation. *In: Winter Conference on Applications of Computer Vision, (WACV)*, IEEE, pp 1328–1336

Fu H, Wang C, Tao D, Black MJ (2016) Occlusion Boundary Detection via Deep Exploration of Context. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 241–250

Fu H, Gong M, Wang C, Batmanghelich K, Tao D (2018) Deep Ordinal Regression Network for Monocular Depth Estimation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 2002–2011

Gaidon A, Wang Q, Cabon Y, Vig E (2016) Virtual Worlds as Proxy for Multi-Object Tracking Analysis. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society

Gan Y, Xu X, Sun W, Lin L (2018) Monocular Depth Estimation with Affinity, Vertical Pooling, and Label Enhancement. *In: European Conference on Computer Vision (ECCV) Part III*, Springer, Lecture Notes in Computer Science, vol 11207, pp 232–247

Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)* 32(11):1231–1237

Geiger D, Ladendorf B, Yuille AL (1995) Occlusions and binocular stereo. *International Journal of Computer Vision (IJCV)* 14(3):211–226

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. *In: International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR.org, JMLR Proceedings, vol 9, pp 249–256

Grammalidis N, Strintzis MG (1998) Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences. *Transactions on Circuits and Systems for Video Technology (TCSVT)* 8(3):328–344

Grard M, Brégier R, Sella F, Dellandréa E, Chen L (2018) Object Segmentation in Depth Maps with One User Click and a Synthetically Trained Fully Convolutional Network. *In: 2017 International Workshop on Human-Friendly Robotics*, Springer Proceedings in Advanced Robotics, vol 7, Springer, pp 207–221

Guan S, Khan AA, Sikdar S, Chitnis PV (2018) Fully Dense UNet for 2D Sparse Photoacoustic Tomography Artifact Removal. *Journal of Biomedical and Health Informatics*

Hayder Z, He X, Salzmann M (2017) Boundary-Aware Instance Segmentation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 587–595

He K, Gkioxari G, Dollár P, Girshick RB (2017) Mask R-CNN. *In: International Conference on Computer Vision (ICCV)*, IEEE Computer Society, pp 2980–2988

He X, Yuille A (2010) Occlusion Boundary Detection Using Pseudo-depth. *In: European Conference on Computer Vision (ECCV) Part IV*, Lecture Notes in Computer Science, vol 6314, Springer, pp 539–552

Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely Connected Convolutional Networks. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 2261–2269

Humayun A, Mac Aodha O, Brostow GJ (2011) Learning to find occlusion regions. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 2161–2168

Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional Architecture for Fast Feature Embedding. *In: International Conference on Multimedia*, ACM, MM'14, pp 675–678

Kendall A, Gal Y, Cipolla R (2018) Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 7482–7491

Kingma DP, Ba J (2015) Adam: A Method for Stochastic Optimization. *In: International Conference on Learning Representations (ICLR)*

Kirillov A, Levinkov E, Andres B, Savchynskyy B, Rother C (2017) InstanceCut: From Edges to Instances with MultiCut. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 7322–7331

Kirillov A, Wu Y, He K, Girshick RB (2019) PointRend: Image Segmentation as Rendering. *CoRR* abs/1912.08193, URL http://arxiv.org/abs/1912.08193, 1912.08193

Kong S, Fowlkes CC (2018) Recurrent Pixel Embedding for Instance Grouping. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 9018–9028

Lee W, Na J, Kim G (2019) Multi-Task Self-Supervised Object Detection via Recycling of Bounding Box An-

notations. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, Computer Vision Foundation / IEEE, pp 4984–4993

Li B, Shen C, Dai Y, van den Hengel A, He M (2015) Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 1119–1127

Li G, Xie Y, Lin L, Yu Y (2017) Instance-Level Salient Object Segmentation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 247–256

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: Common Objects in Context. *In: European Conference on Computer Vision (ECCV) Part V*, Springer, Lecture Notes in Computer Science, vol 8693, pp 740–755

Lin TY, Goyal P, Girshick RB, He K, Dollár P (2017) Focal Loss for Dense Object Detection. *In: International Conference on Computer Vision (ICCV)*, IEEE Computer Society, pp 2999–3007

Liu F, Shen C, Lin G, Reid ID (2016) Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. *IEEE Transactions on Pattern Analysis Machine Intelligence (TPAMI)* 38(10):2024–2039

Liu G, Si J, Hu Y, Li S (2018a) Photographic image synthesis with improved U-net. *In: International Conference on Advanced Computational Intelligence (ICACI)*, IEEE, pp 402–407

Liu R, Lehman J, Molino P, Such FP, Frank E, Sergeev A, Yosinski J (2018b) An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution. *In: Advances in Neural Information Processing Systems (NeurIPS)*, pp 9628–9639

Liu S, Qi L, Qin H, Shi J, Jia J (2018c) Path Aggregation Network for Instance Segmentation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 8759–8768

Liu S, Johns E, Davison AJ (2019) End-to-End Multi-Task Learning with Attention. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, Computer Vision Foundation / IEEE, pp 1871–1880

Liu Y, Cheng MM, Hu X, Wang K, Bai X (2017) Richer Convolutional Features for Edge Detection. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 5872—5881

Luo P, Wang G, Lin L, Wang X (2017) Deep Dual Learning for Semantic Image Segmentation. *In: International Conference on Computer Vision (ICCV)*, IEEE Computer Society, pp 2737–2745

Maninis KK, Pont-Tuset J, Arbeláez PA, Gool LJV (2016) Convolutional Oriented Boundaries. *In: European Conference on Computer Vision (ECCV) Part I*, Springer, Lecture Notes in Computer Science, vol 9905, pp 580–596

Martin D, Fowlkes C, Tal D, Malik J (2001) A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *In: International Conference on Computer Vision (ICCV)*, IEEE Computer Society, pp 416–423

McCormac J, Handa A, Leutenegger S, Davison AJ (2017) SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation? *In: International Conference on Computer Vision (ICCV)*, IEEE Computer Society, pp 2697–2706

Misra I, Shrivastava A, Gupta A, Hebert M (2016) Cross-Stitch Networks for Multi-task Learning. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 3994–4003

Novotný D, Albanie S, Larlus D, Vedaldi A (2018) Semi-convolutional Operators for Instance Segmentation. *In: European Conference on Computer Vision (ECCV) Part I*, Springer, Lecture Notes in Computer Science, vol 11205, pp 89–105

Pont-Tuset J, Arbelaez P, Barron JT, Marqus F, Malik J (2017) Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39(1):128–140

Qi L, Jiang L, Liu S, Shen X, Jia J (2019) Amodal Instance Segmentation With KINS Dataset. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, Computer Vision Foundation / IEEE, pp 3014–3023

Ren M, Zemel RS (2017) End-to-End Instance Segmentation with Recurrent Attention. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 293–301

Ren X, Fowlkes CC, Malik J (2006) Figure/Ground Assignment in Natural Images. *In: European Conference on Computer Vision (ECCV) Part II*, Springer, Lecture Notes in Computer Science, vol 3952, pp 614–627

Romera-Paredes B, Torr PHS (2016) Recurrent Instance Segmentation. *In: European Conference on Computer Vision (ECCV) Part VI*, Springer, Lecture Notes in Computer Science, vol 9910, pp 312–329

Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation, Springer, pp 234–241. Lecture Notes in Computer Science

Ros G, Sellart L, Materzynska J, Vázquez D, López AM (2016) The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 3234–3243

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3):211–252

Shi W, Caballero J, Huszar F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 1874–1883

Simonyan K, Zisserman A (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. *In: International Conference on Learning Representations (ICLR)*, IEEE Computer Society

Stein A, Hebert M (2006) Local Detection of Occlusion Boundaries in Video. *In: British Machine Vision Conference (BMVC)*

Sun D, Liu C, Pfister H (2014) Local Layering for Joint Motion Estimation and Occlusion Detection. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 1098–1105

Tang Z, Peng X, Geng S, Wu L, Zhang S, Metaxas DN (2018) Quantized Densely Connected U-Nets for Efficient Landmark Localization. *In: European Conference on Computer Vision (ECCV) Part III*, Springer, Lecture Notes in Computer Science, vol 11207, pp 348–364

Wang G, Wang X, Li FWB, Liang X (2018a) DOOBNet: Deep Object Occlusion Boundary Detection from an Image. *In: Asian Conference on Computer Vision (ACCV) Part VI*, Springer, Lecture Notes in Computer Science, vol 11366, pp 686–702

Wang P, Yuille AL (2016) DOC: Deep OCclusion Estimation from a Single Image. *In: European Conference on Computer Vision (ECCV) Part I*, Springer, Lecture Notes in Computer Science, vol 9905, pp 545–561

Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, Cottrell GW (2018b) Understanding Convolution for Semantic Segmentation. *In: Winter Conference on Applications of Computer Vision (WACV)*, pp 1451–1460

Wang Y, Zhao X, Huang K (2017) Deep Crisp Boundaries. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 1724–1732

Williams O, Isard M, MacCormick J (2011) Estimating Disparity and Occlusions in Stereo Video Sequences. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 250–257

Xie S, Tu Z (2015) Holistically-Nested Edge Detection. *In: International Conference on Computer Vision (ICCV)*, IEEE Computer Society, pp 1395–1403

Yang J, Price BL, Cohen S, Lee H, Yang MH (2016) Object Contour Detection with a Fully Convolutional Encoder-Decoder Network. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 193–202

Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? *In: Advances in Neural Information Processing Systems (NIPS)*, pp 3320–3328

Yu F, Koltun V (2016) Multi-Scale Context Aggregation by Dilated Convolutions. *In: International Conference on Learning Representations (ICLR)*

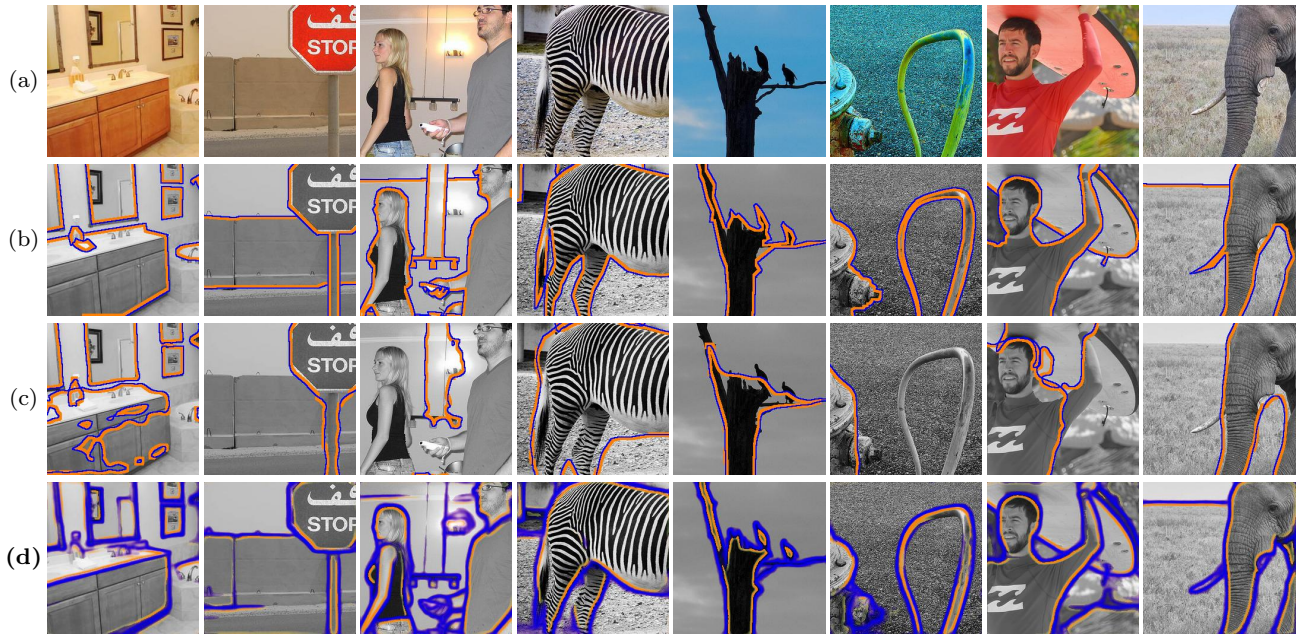
Yu J, Yang L, Xu N, Yang J, Huang T (2019) Slimmable Neural Networks. *In: International Conference on Learning Representations (ICLR)*

Yu Z, Liu W, Zou Y, Feng C, Ramalingam S, Kumar BVKV, Kautz J (2018) Simultaneous Edge Alignment and Learning. *In: European Conference on Computer Vision (ECCV) Part III*, Springer, Lecture Notes in Computer Science, vol 11207, pp 400–417

Zhang L, Li X, Arnab A, Yang K, Tong Y, Torr PH (2019) Dual Graph Convolutional Network for Semantic Segmentation. *In: British Machine Vision Conference (BMVC)*

Zhu Y, Tian Y, Metaxas DN, Dollár P (2017) Semantic Amodal Segmentation. *In: Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp 3001–3009

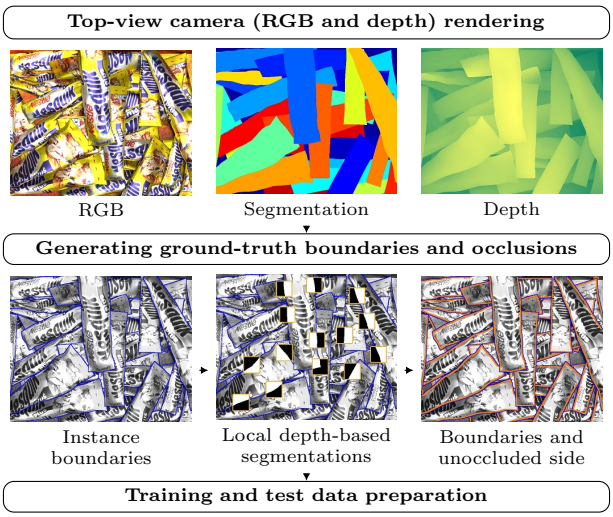
Zitnick CL, Kanade T (2000) A Cooperative Algorithm for Stereo Matching and Occlusion Detection. *IEEE Transactions on Pattern Analysis Machine Intelligence (TPAMI)* 22(7):675–684



Approach	*All regions*				*Things*[1] *only*				*Stuff*[1] *only*			
	Boundaries		**Occlusions**		**Boundaries**		**Occlusions**		**Boundaries**		**Occlusions**	
	ODS	AP	ODS	AP	ODS	AP	ODS	AP	ODS	AP	ODS	AP
(c) Amodal segmentation [2]	.492	–	.529	–	.536	–	.608	–	.489	–	.397	–
(d) **MC3† (Ours)**	**.666**	.694	**.637**	.673	**.666**	.690	**.640**	.674	**.687**	.727	**.648**	.693

[1] Things are objects with well-defined shape, *e.g.* car, person, and stuff instances amorphous regions, *e.g.* grass, sky (Caesar et al, 2018).
[2] The evaluation is performed on the binary segment proposals made available by the authors. We derive occlusion-aware boundaries from the ground truth and the precomputed results alike: after intersecting the modal and amodal masks of an instance, the amodal pixels that don't belong to the intersection are considered as occluded.

Fig. 15: Comparative results for instance boundary (blue) and unoccluded boundary side (orange) detection on COCOA. From top to bottom: input (a), ground truth (b), inference by amodal instance segmentation (Zhu et al, 2017) (c), using a bicameral structure (d). Unlike the proposed approach, using a region proposal-based detection qualitatively leads to coarse segmentations and non-detected instances. Best viewed in color.

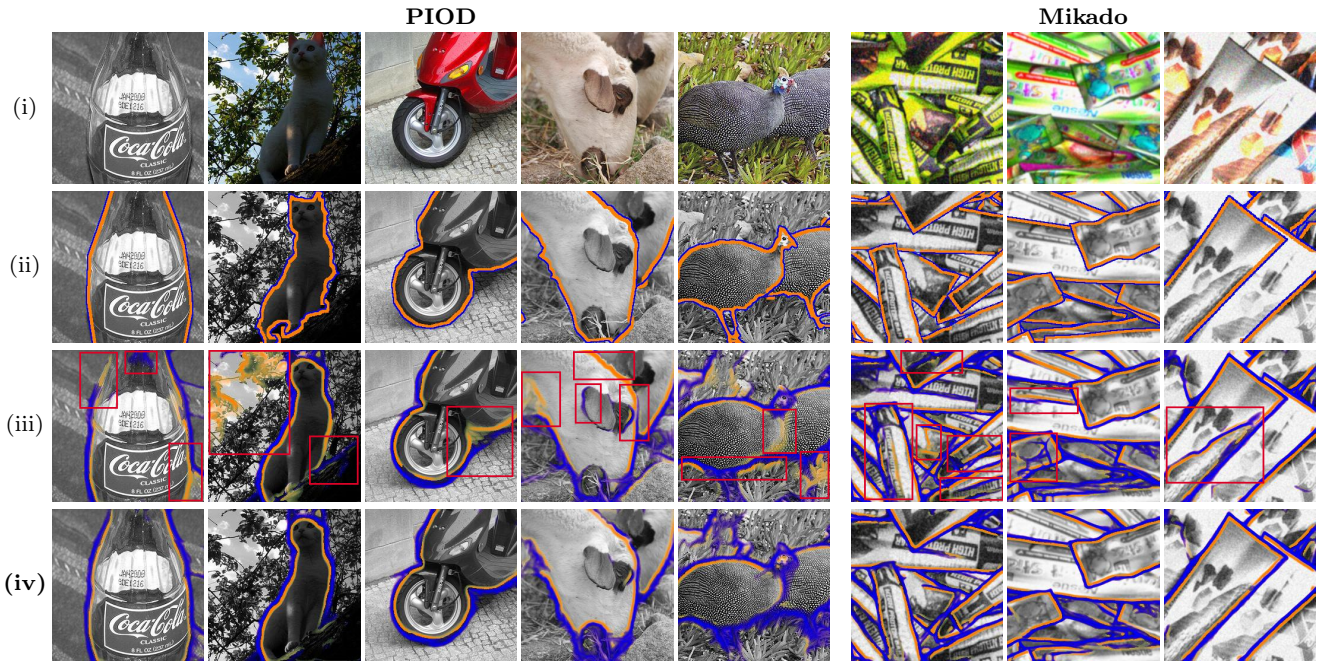


(a) Pipeline for generating the ground-truth occluding boundary side. At each boundary pixel, a depth-based binary segmentation of the neighborhood is performed to label each side, such that the higher side is set to 1 and the lower side to 0.

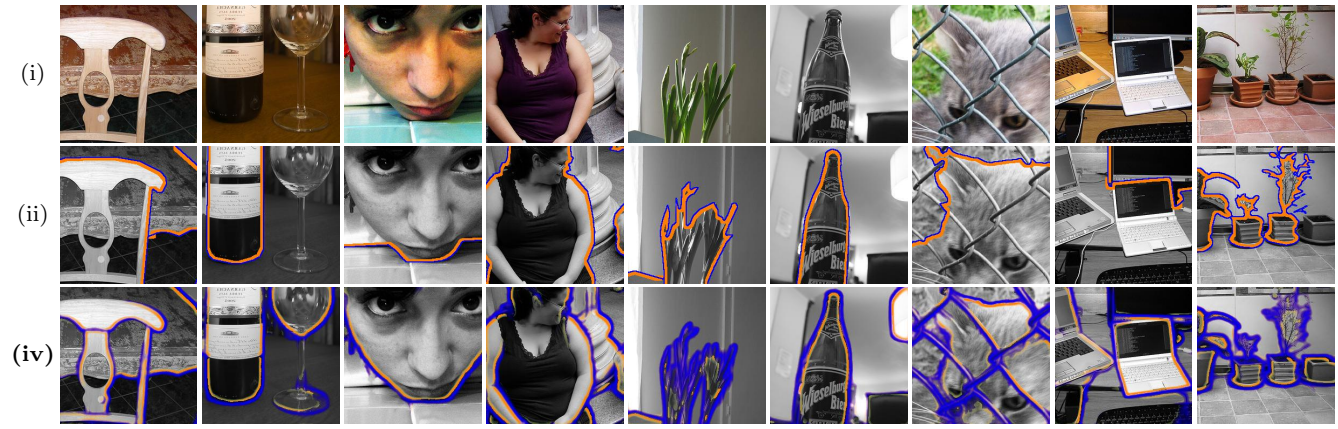
(a) Overview of the sachet textures used for generating Mikado.

(b) Overview of the background textures used for generating Mikado.

Fig. 16: Supplementary material on the proposed synthetic data generation pipeline.



(a) From top to bottom: input (i), ground truth (ii), inference using two independent streams (iii), using a bicameral structure (iv). Instance boundaries are in blue, their unoccluded side in orange. Red rectangles highlight some false positive erased when using instead a single encoder shared by cascaded decoders.



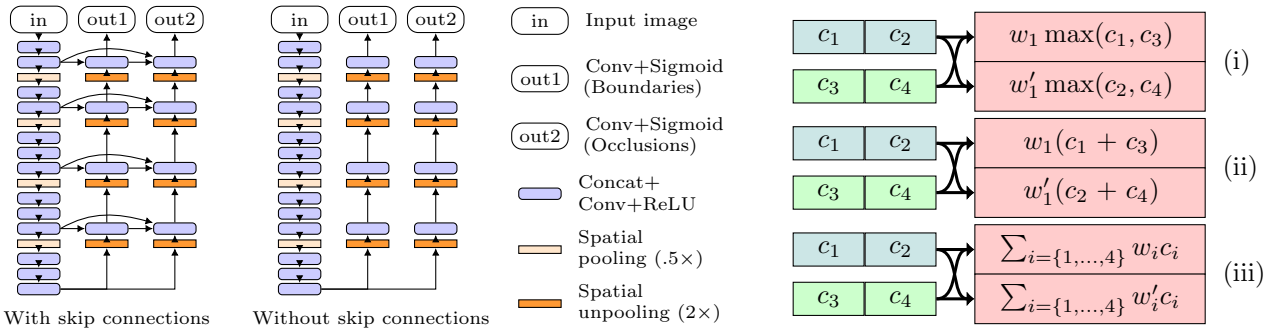
(b) From top to bottom: input (i), ground-truth (ii), inference using a bicameral structure (iv). The proposed network fairly infers non-annotated boundaries and delineates instances coarsely annotated by humans.

Trained on	Tests on **Mikado**			
	Boundaries		**Occlusions**	
	ODS	AP	ODS	AP
Mikado	.769	.847	.801	.884
PIOD	.300	.233	.326	.267

Trained on	Tests on **PIOD**			
	Boundaries		**Occlusions**	
	ODS	AP	ODS	AP
PIOD	.697	.738	.692	.747
Mikado	.405	.350	.400	.349

(c) Cross-dataset performances between Mikado and PIOD using a bicameral design. Both datasets perform poorly on each other because they follow very different texture, shape, and pose distributions.

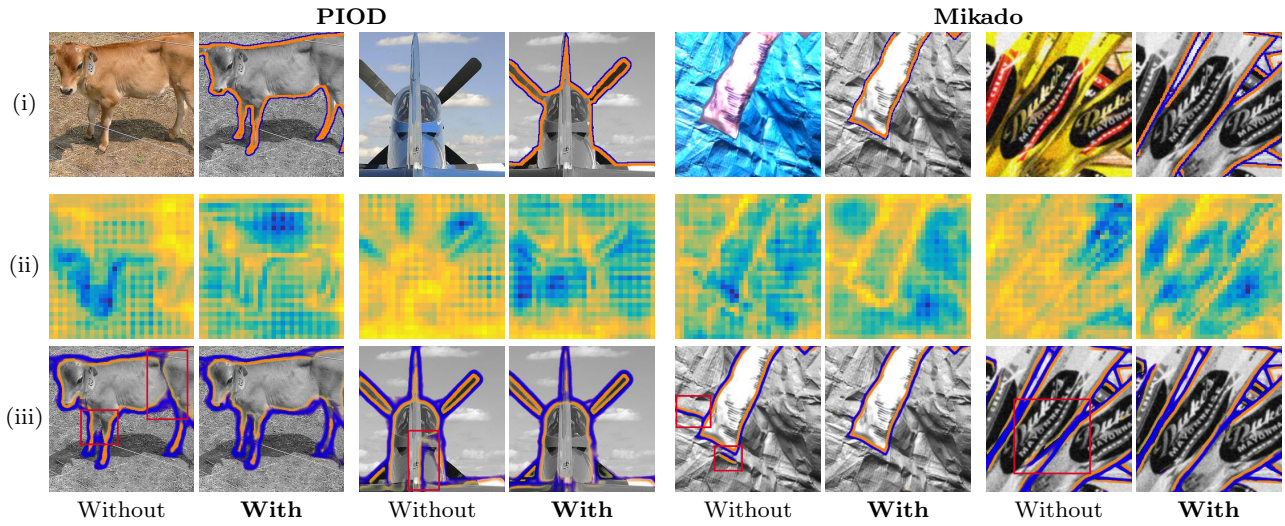
Fig. 17: Comparative results for occlusion-aware boundary detection on PIOD and Mikado. Best viewed in color.



(a) Left: a bicameral structure with and without skip connections. Right: different skip connection types for merging two 2-channel feature vectors (c_1, c_2) and (c_3, c_4) into a new 2-channel one, using parameters w_i and w'_i. From top to bottom: by element-wise max (i); by element-wise sum (ii); by concatenation (iii).

Dataset:	PIOD						Mikado					
Skip connections?	**Boundaries**			**Occlusions**			**Boundaries**			**Occlusions**		
(Type)	ODS	AP	AP_{60}	ODS	AP	AP_{60}	ODS	AP	AP_{60}	ODS	AP	AP_{60}
No	.693	**.744**	.495	.692	**.749**	.520	.759	.834	.686	.793	.878	.748
Yes (Element-wise max)	.685	.729	.512	.676	.731	.522	.755	.830	.676	.786	.871	.735
Yes (Element-wise sum)	.687	.730	.505	.678	.731	.514	.761	.838	.685	.791	.876	.743
Yes (Concatenation)	**.697**	.738	**.517**	**.692**	.747	**.532**	**.769**	**.847**	**.698**	**.801**	**.884**	**.758**

(b) Comparative performances on PIOD and Mikado.



(c) From top to bottom: input and ground truth (i), activation map after the affine transformation on top of the first unpooling layer of the boundary branch (ii), inference (iii). Combining spatial information and higher-level semantics at each scale using skip connections between the encoder and decoders enables to detect instance boundaries earlier when decoding.

Architecture	**Encoder backbone**	**Number of parameters**	**Boundaries**			**Occlusions**		
			ODS	AP	AP_{60}	ODS	AP	AP_{60}
Two streams (Baseline)	VGG16	46,839,938 (×1.0)	.673	.708	.476	.681	.733	.518
Bicameral decoder	VGG16	34,301,250 (×.73)	.697	.738	.517	.692	.747	.532
	DenseNet121	33,009,846 (×.70)	**.712**	**.761**	**.529**	**.714**	**.778**	**.556**

(d) Plugging a bicameral decoder to a deeper encoder with DenseNet blocks (Huang et al, 2017) enables to capture better contextual representations of the image, thus improving the detection of occlusion-aware boundaries.

Fig. 18: Comparative results for occlusion-aware boundary detection on PIOD and Mikado, using a bicameral structure: with and without skip connections, with different types of skip connections, with different encoder backbones. Best viewed in color.

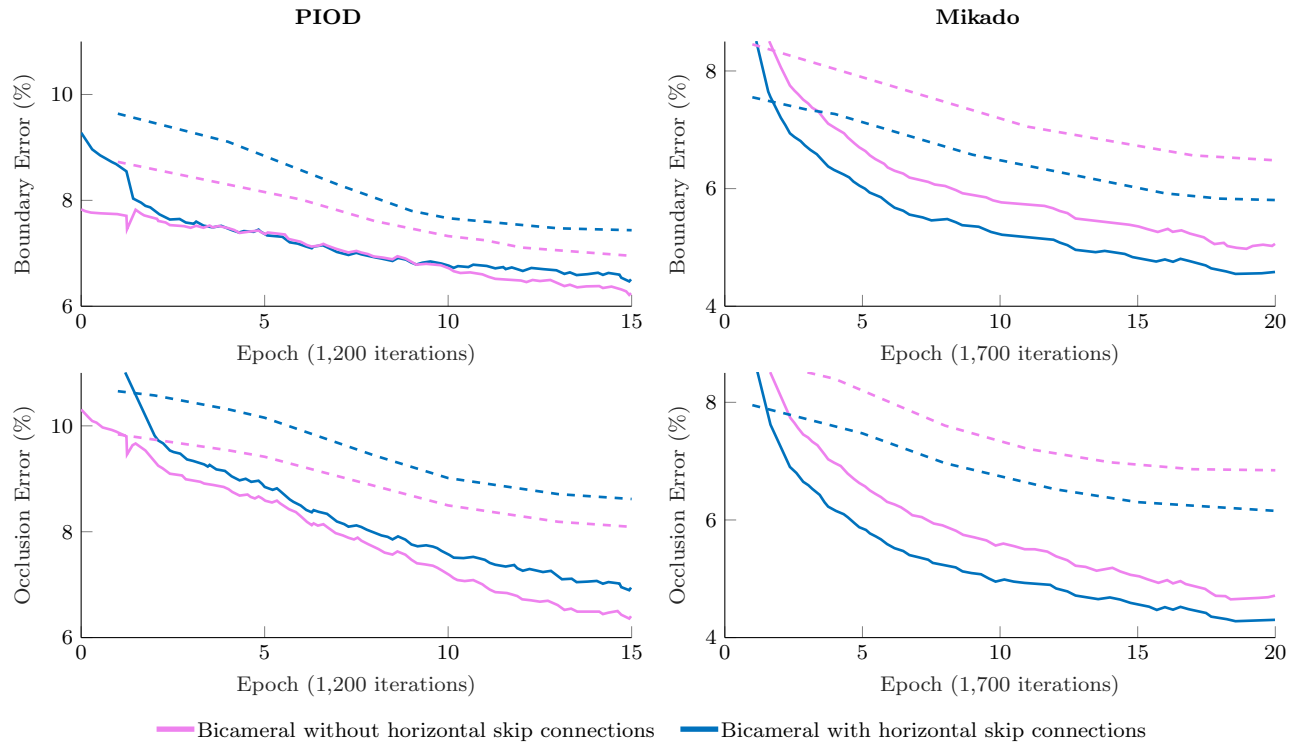
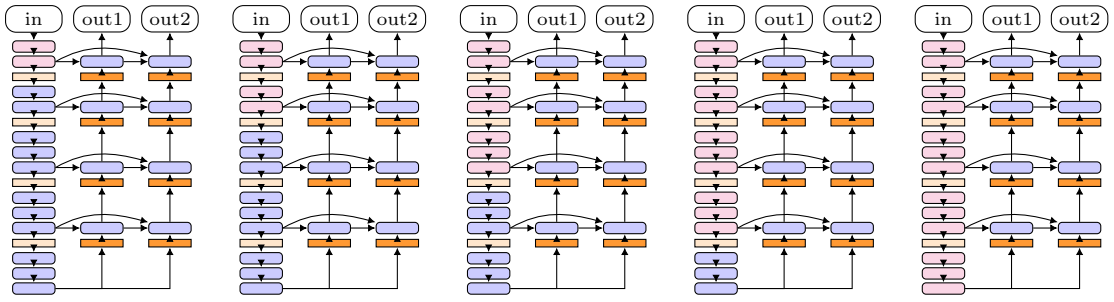


Fig. 19: Training (solid) and test (dashed) errors for instance boundary (top) and occluding boundary side (bottom) detection on PIOD (left) and Mikado (right) using different network architectures. Lower boundary and occlusion errors are reached when jointly learning boundaries and occlusions (green, blue, yellow, purple) rather than independently (red). Best viewed in color.



Pretraining on Mikado	Finetuning on D2SA	Number of real training images	Frozen encoder blocks* (in pink)	Boundaries ODS	Boundaries AP	Occlusions ODS	Occlusions AP
No	Yes	438	None	.700	.715	.725	.756
		Augmented (D2SA+)		.783	.792	.785	.795
Yes	No	0	–	.652	.649	.458	.400
	Yes	438	None	.780	.808	.794	.830
			1	.783	.803	.797	.829
			1, 2	.780	.802	.793	.827
			1, 2, 3	**.793**	**.819**	**.810**	**.849**
			1, 2, 3, 4	.759	.799	.769	.819
			1, 2, 3, 4, 5	.767	.815	.773	.823

* A block is a set of convolutional layers between two pooling layers; a VGG16-based encoder is therefore composed of 5 blocks.

Fig. 20: Comparative performances of a bicameral structure on D2SA using different pretraining conditions. Performances on both boundaries and occlusions are maximized when freezing at finetuning time the first three encoder blocks pretrained on Mikado. Best viewed in color.