International Journal of Computer Vision
https://doi.org/10.1007/s11263-021-01430-6

DeepFlux for Skeleton Detection in the Wild

Yongchao Xu[1] · Yukang Wang[2] · Stavros Tsogkas[3,5] · Jianqiang Wan[2] · Xiang Bai[2] · Sven Dickinson[3,4,5] · Kaleem Siddiqi[6]

Received: 5 June 2020 / Accepted: 4 January 2021

Abstract
The medial axis, or skeleton, is a fundamental object representation that has been extensively used in shape recognition. Yet, its extension to natural images has been challenging due to the large appearance and scale variations of objects and complex background clutter that appear in this setting. In contrast to recent methods that address skeleton extraction as a binary pixel classification problem, in this article we present an alternative formulation for skeleton detection. We follow the spirit of flux-based algorithms for medial axis recovery by training a convolutional neural network to predict a two-dimensional vector field encoding the flux representation. The skeleton is then recovered from the flux representation, which captures the position of skeletal pixels relative to semantically meaningful entities (e.g., image points in spatial context, and hence the implied object boundaries), resulting in precise skeleton detection. Moreover, since the flux representation is a region-based vector field, it is better able to cope with object parts of large width. We evaluate the proposed method, termed DeepFlux, on six benchmark datasets, consistently achieving superior performance over state-of-the-art methods. Finally, we demonstrate an application of DeepFlux, augmented with a skeleton scale estimation module, to detect objects in aerial images. This combination yields results that are competitive with models trained specifically for object detection, showcasing the versatility and effectiveness of mid-level representations in high-level tasks. An implementation of our method is available at https://github.com/YukangWang/DeepFlux.

Keywords Skeleton detection · Medial axis · Flux representation · Convolutional neural network · Mid-level representation

1 Introduction

The shape skeleton, or medial axis [4], is a structure-based object descriptor that reveals local symmetry as well as connectivity between object parts [11,39]. Modeling objects via their axes of symmetry and, in particular, using skeletons has a long history in computer vision. Skeletonization algorithms provide a concise and effective representation of deformable objects, while supporting many applications, including object recognition and retrieval [3,17,58,73], pose

Communicated by Christoph H. Lampert.

Disclaimer: Sven Dickinson and Stavros Tsogkas contributed to this article in their personal capacity as Professor and Adjunct Professor, respectively, at the University of Toronto. The views expressed (or the conclusions reached) are their own and do not necessarily represent the views of Samsung Research America, Inc.

✉ Xiang Bai
xbai@hust.edu.cn

Yongchao Xu
yongchao.xu@whu.edu.cn

Yukang Wang
wangyk@hust.edu.cn

Stavros Tsogkas
tsogkas@cs.toronto.edu

Jianqiang Wan
jianqw@hust.edu.cn

Sven Dickinson
sven@cs.toronto.edu

Kaleem Siddiqi
siddiqi@cim.mcgill.ca

[1] School of Computer Science, Wuhan University, Wuhan, China

[2] School of EiC, Huazhong University of Science and Technology, Wuhan, China

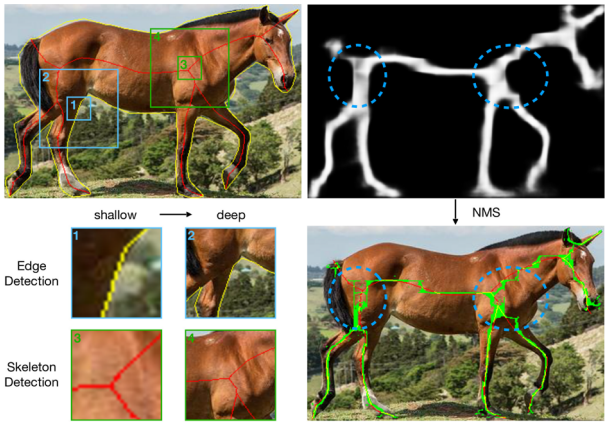
[3] University of Toronto, Toronto, Canada

[4] Vector Institute for Artificial Intelligence, Toronto, Canada

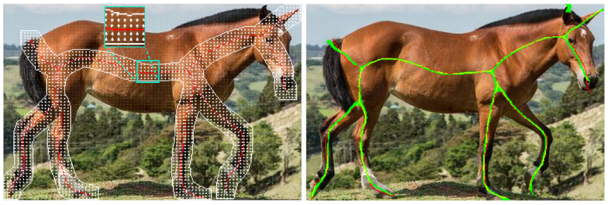
[5] Samsung Toronto AI Research Center, Toronto, Canada

[6] School of Computer Science and Centre for Intelligent Machines, McGill University, Montreal, Canada

Published online: 30 January 2021



(a) Previous CNN-based skeleton detection methods rely on NMS to obtain their final results.



(b) A flux-based representation models the local spatial context of skeletal points, allowing for precise detection.

Fig. 1 **a** Previous CNN-based methods treat skeleton detection as binary pixel classification, followed by non-maximum suppression (NMS). This can result in poor localization as well as disconnected segments. **b** The proposed DeepFlux method models the spatial context of skeletal points using a novel flux representation (left). The flux vector field encodes the position of skeletal points in relation to their associated image pixels, and hence also the implied object boundaries. This allows one to associate skeletal pixels with sinks, where the flux is absorbed, in the spirit of flux-based skeletonization methods [52]. Red: ground truth skeleton; Green: detected skeleton (Color figure online)

estimation [19,51,62], hand gesture recognition [46], shape matching [54], scene text detection [71], and road detection in aerial scenes [42,43,57].

Early algorithms for computing skeletons directly from images [21,29,30,32,44,69,70] yield a gradient intensity map, driven by geometric constraints between skeletal pixels and edge fragments. Such methods cannot easily handle complex image data without prior information about object shape and location. Learning-based methods [28,48,55,57,60], on the other hand, demonstrate an improved ability for object skeleton detection in natural images, but are still unable to cope with complex backgrounds or clutter.

Convolutional neural networks (CNNs) are a specific instance of learning frameworks that have led to vast improvements in the performance of object skeleton detection algorithms in recent years [25,31,33,49,50,66,72]. CNN-based methods typically frame the problem as one of binary pixel classification: given a dataset of images containing objects, paired with their (binary) skeleton annotations, the network is trained to predict the probability of each pixel belonging to a skeleton. The ground truth skeletons are usually extracted by applying a binary skeletonization algorithm to pre-segmented masks of the objects present in the image. As a result, the skeletons detected by each model are dataset-dependent. For instance, some datasets may only contain skeleton annotations for a single foreground object [47], while others may involve scenes with multiple objects [25], or may include annotations for background structures [60]. This stands in contrast to recent work in unsupervised medial axis extraction from natural scenes [15,59].

Most of the aforementioned CNN-based methods derive from the Holistically-Nested Edge Detection (HED) model [65] or variations of it that better leverage multi-level features for capturing skeletons across a range of spatial scales. However, object skeleton computation in natural images using CNNs is inherently different from the problem of edge detection. As illustrated in Fig. 1a, edges associated with object boundaries can often be detected using information such as local appearance or texture changes. Such cues can be picked up by the more spatially accurate, shallow convolutional layers. Object skeletons, however, embody medial properties and high-level semantics. They are situated at regions within object parts that exhibit *local bilateral symmetry*, since the medial axis bisects the object angle [53]. Successfully detecting skeletons purely from local image information (*e.g.*, the green box numbered 3 in Fig. 1a) is challenging, since this requires reasoning over a larger spatial extent, such as the width of the torso of the horse in this case. Layers deeper in the CNN architecture are more appropriate for computing features at such coarser scales, but this presents a confound. Coarse features might not provide accurate spatial localization of the object skeleton.

In this paper, we propose a novel notion of *spatial context flux*, to accurately detect object skeletons within a CNN framework. Models based on the related notion of a field potential have also shown promise for other visual processing tasks that require non-local interactions, such as border ownership computations in the visual cortex [74]. We start by considering the spatial context of the skeleton, *i.e.*, a neighborhood around a skeleton branch. For each context pixel, we define a two-dimensional unit flux vector pointing to the nearest skeleton pixel, generating a flux vector field. Within this representation, the object skeleton corresponds to pixels where the net inward flux is positive, following the motivation behind past flux-based methods for skeletonizing binary objects [12,52]. We then use a CNN to learn the spatial context flux, via a pixel-wise regression task in place of binary classification. The learned flux vector field encodes the relative locations of context and skeleton pixels, enabling the accurate recovery of the object skeleton via a simple post-processing step. Explicitly leveraging skeleton spatial context in our representation provides a larger receptive field size for estimation. This is helpful both for

detecting medial points associated with larger spatial scales, and for more robust localization around junctions.

The present article builds upon work first presented in [61]. Our contributions can be enumerated as follows.

1. We propose a novel *spatial context flux* representation for object skeleton detection. This concept explicitly encodes the relationship between image pixels and their closest skeletal points.
2. Using this spatial context flux, we develop a method which we dub *DeepFlux*, that accurately and efficiently detects object skeletons in an image.
3. DeepFlux consistently outperforms state-of-the-art methods on six public benchmarks. To our knowledge, this is the first application of flux concepts, which have been successfully used for skeletonization of binary objects [12,52], to the detection of object skeletons in natural images. It is also the first attempt at learning such flux-based representations directly from natural images.

A preliminary version of this study was presented in [61]. The current journal extension introduces two major improvements. First, we replace the post-processing step with a convolutional module, making the pipeline trainable in an end-to-end manner, while improving performance, and requiring less runtime. Second, in addition to the flux and skeleton branch, we also learn the associated skeleton scale for DeepFlux to detect objects in aerial images, achieving competitive performance against classical CNN-based object detectors.

2 Related Work

Object skeletonization has been widely studied in the last few decades. In our review, we contrast traditional, bottom-up methods, with those that rely on supervised learning on annotated skeleton datasets.

2.1 Bottom-Up Skeletonization Methods

Many early skeleton detection algorithms [21,29,30,32,44,69,70] are based on gradient intensity maps. In [52], the authors study the limiting average outward flux of the gradient of a Euclidean distance function to a 2D or 3D object boundary. The skeleton is associated with those locations where an energy principle is violated, where there is a net inward flux. Other researchers have constructed the skeleton by merging local skeleton segments with a learned segment-linking model. Levinshtein et al. [28] propose a method to work directly on images, which uses multi-scale super-pixels and a learned affinity between adjacent super-pixels to group proximal medial points. A graph-based clustering algorithm is then applied to form the complete skeleton. Lee et al. [55] improve the approach in [28] by using a deformable disc model, which can detect curved and tapered symmetric parts. A novel definition of an appearance medial axis transform (AMAT) has been proposed in [59], to detect symmetry in the wild in a purely bottom-up, unsupervised fashion. In recent follow-up work [15], the AMAT framework is augmented by explicitly incorporating rules from the Shock Grammar for shapes [54], resulting in significant improvements in computational speed and medial axis quality. Finally, [22] describes an interesting framework for solving segmentation and skeletonization by exploiting the commonalities among different images of semantically similar objects, in a joint co-segmentation and co-skeletonization optimization scheme.

2.2 Learning-Based Skeleton Detection

In more recent literature [48,57,60], object skeleton detection is treated as a pixel-wise classification or regression problem, and is solved using supervised learning. Tsogkas and Kokkinos [60] extract hand-designed features at each pixel and train a classifier for symmetry detection. They employ a multiple instance learning (MIL) framework to accommodate the unknown scale and orientation of symmetry axes. Shen et al. [48] extend the approach in [60] by training a group of MIL classifiers to capture the diversity of symmetry patterns. Sironi et al. [57] propose a regression-based approach to improve the accuracy of skeleton locations. They train regressors which learn the distances to the closest skeleton in scale-space and identify the skeleton by finding the local maxima.

With the popularization of CNNs, deep learning-based methods [25,31,33,49,50,72] have shown great promise for object skeleton detection. Shen et al. [50] propose an approach which fuses scale-associated deep side-outputs (FSDS), based on the architecture of HED [65]. Since skeletons at different spatial scales can be captured in different stages, they supervise the side outputs with scale-associated ground-truth data. They then extend their original method by learning multi-task scale-associated deep side outputs (LMSDS) in [49].

This leads to improved skeleton localization and scale prediction, and better overall performance. Ke et al. [25] present a side-output residual network (SRN), which leverages the output residual units to fit the errors between the ground-truth and the side-outputs. By cascading residual units in a deep-to-shallow manner, SRN can effectively detect the skeleton at different scales. Liu et at. [33] develop a two-stream network that combines image and segmentation cues to capture complementary information for skeleton localization. Zhao et al. introduce a hierarchical feature integration (Hi-Fi) mechanism in [72], where multi-scale features are integrated with bidirectional guidance so that high-level semantics and low-

level details can benefit from each other. Liu et al. [31] propose a linear span network (LSN) that uses linear span units to increase the independence of convolutional features and the efficiency of feature integration. In [66], Xu et al. introduce a geometry-aware objective function based on Hausdorff distance, to better incorporate geometric constraints.

2.3 Features of DeepFlux

Though the method we propose in the present paper also benefits from CNN-based learning, it differs from the methods in [25,31,33,49,50,72] in a fundamental way, due to its different learning objective. Instead of treating object skeleton detection in natural images as a binary classification problem, DeepFlux focuses on learning the spatial context flux of skeletons, and as such includes more informative non-local cues, such as the relative position of skeleton points to image points in their vicinity. Thus, the relationship between skeletal point locations and their associated object boundaries is also captured, at least implicity. A direct consequence of this powerful image context flux representation is that a simple post-processing step can recover the skeleton directly from the learned flux. In this manner, we avoid the inaccurate localization of skeletal points by non-maximum suppression used in previous deep learning methods. In addition, DeepFlux enlarges the spatial extent used by the CNN to detect the skeleton, through its use of spatial context flux. This region-based flux representation allows our approach to capture larger object parts.

We note that the proposed DeepFlux is similar in spirit to the original notion of flux [12,52] that is defined based on an object boundary, for skeletonization of 2D/3D binary objects. As such, DeepFlux inherits its mathematical properties including the unique mapping of skeletal points to boundary points. However, the present article is the first to extend this notion of flux to skeleton detection in natural images, where the flux is computed on dilated skeletons in a supervised learning setting. Our work is also related to the approaches in [1,2,6,9,27,38,45,67] which learn direction cues for edge detection, instance segmentation, and pose estimation. In the present article, this direction information is encoded in the flux representation, and is implicitly learned for skeleton recovery.

2.4 Direction Fields in Models of Spatial Context

The use of direction fields to model spatial context has also shown promise in other computer vision applications, including image segmentation, object segmentation, and pose estimation. In [38], the authors propose to learn edge directions in addition to edge location, for generic image segmentation. Other methods make use of a direction field defined on regions of interest to achieve instance segmentation, such as the deep watershed transform in [2], which regresses the distance map to boundaries obtained by semantic segmentation. A similar direction field on text areas is proposed in [67], to extract instances of text in scenes, whereas direction cues pointing to object centers are used to improve instance and video segmentation in [6] and [9], respectively. Finally, direction cues are also used to improve instance segmentation in [1] and direction fields pointing towards keypoints are used for pose estimation in [27,45].

3 Method

3.1 Overview

Many recent CNN-based skeleton detection approaches build on some variant of the HED architecture [65]. The combination of a powerful classifier (CNN) and the use of side outputs to extract and combine features at multiple scales has enabled these systems to accurately localize medial points of objects in natural images. However, while state-of-the-art skeleton detection systems are quite effective at extracting medial axes of elongated structures, they still struggle when reasoning about ligature areas. This is not a surprise, because in contrast to the skeletal branches they connect, ligature areas exhibit much less structural regularity, making their exact localization ambiguous. As a result, most methods result in poor localization of ligature points, or fragmentation of medial axis segments between the medial axes representing object parts.

We propose to mitigate this problem by casting skeleton detection as the problem of predicting a two-dimensional flux field from scene points to nearby skeleton points, within a fixed-size neighborhood. We then define skeleton points as the local flux minima, or, alternatively, as sinks "absorbing" flux from nearby points. We argue –and show empirically in our experiments– that this approach leads to more robust localization and better connectivity between skeletal branches. We also argue that considering a small neighborhood around the true skeleton points is sufficient, consistent with past approaches to binary object skeletonization [12]. Whereas predicting the flux for the entire object would allow us to also infer the medial radius function, in this work we focus on improving medial point localization, and employ existing ideas for integrating scale prediction into our network, to tackle a high-level task in Sect. 5. The overall pipeline of the proposed method, which we dub *DeepFlux*, is depicted in Fig. 2.

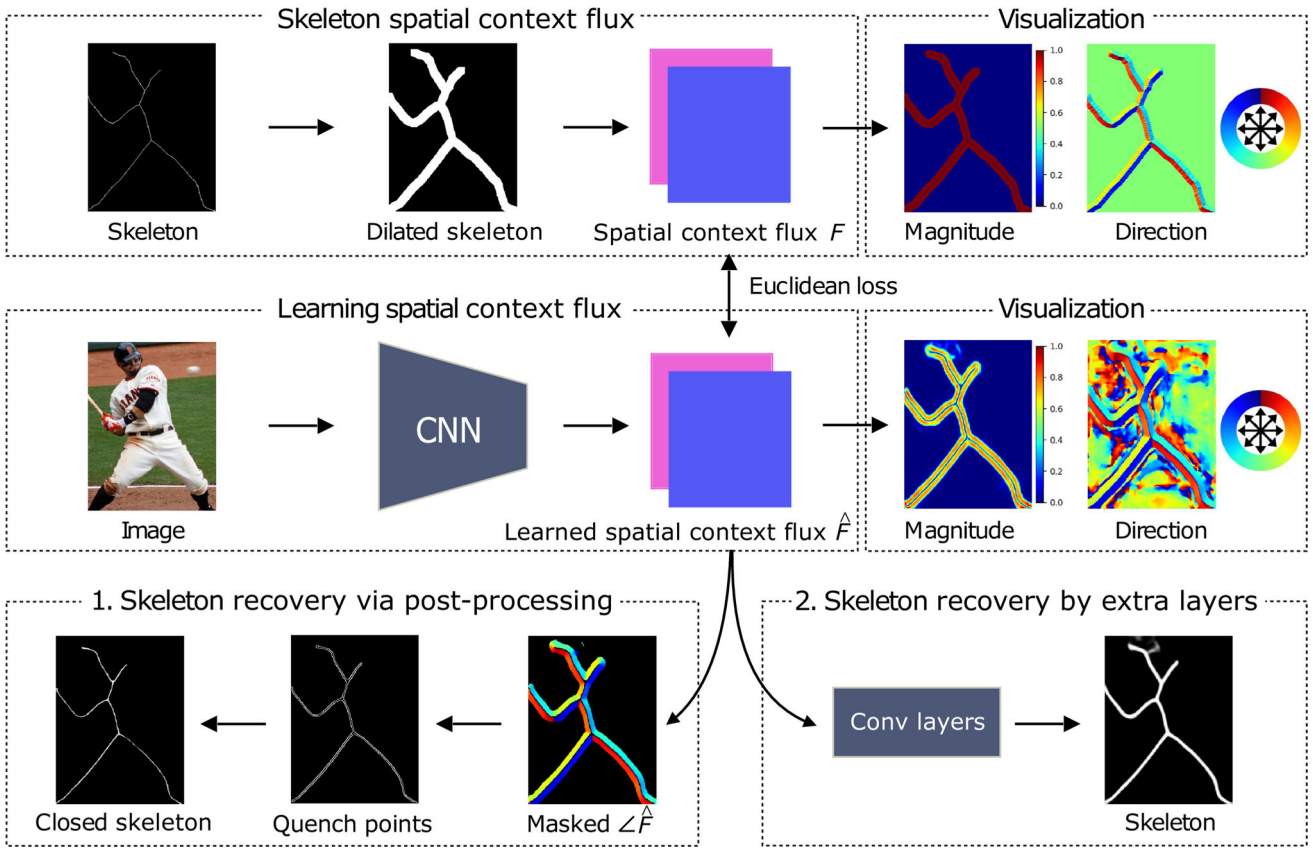


Fig. 2 The pipeline of the proposed method. For an input image, the network computes a two-dimensional vector field of symmetry spatial context flux (with a visualization of its magnitude and direction on the right). Based on this flux representation, we can recover medial axes reflecting object part symmetries by localizing points with high inward flux (followed by a morphological closing), or by using additional convolution layers, which makes the entire pipeline end-to-end trainable

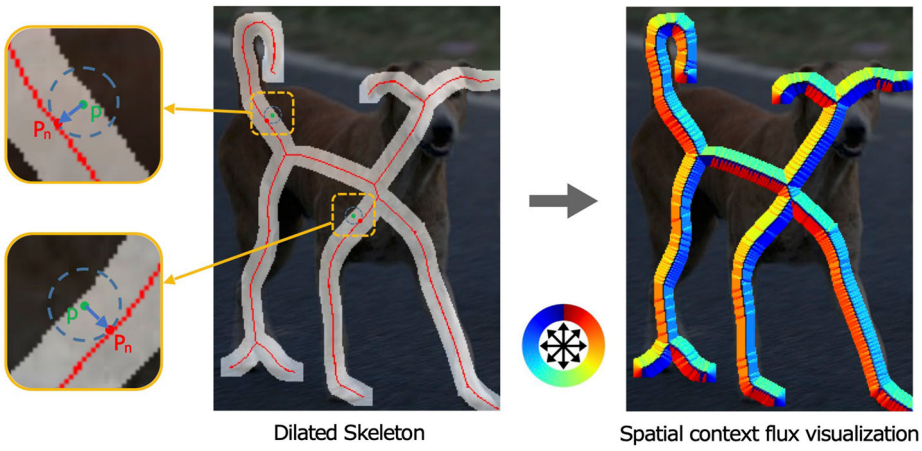


Fig. 3 Dilating the object skeleton with a fixed-size disk defines a neighborhood of "skeleton spatial context", shown in the middle figure as a binary mask. For each pixel $\mathbf{p}$ within this neighborhood (excluding skeleton points), let p_n be its nearest skeleton point. The flux $\mathbf{F}(\mathbf{p})$ is defined as the two-dimensional unit vector pointing away from $\mathbf{p}$ to p_n. In the figure on the right, we visualize the orientation of all points in the skeleton neighborhood (the color wheel shows the correspondence between color and orientation). For the pixels outside the skeleton context neighborhood, there is no flux, *i.e.*, the flux is set to $(0, 0)$ (Color figure online)

3.2 Spatial Context Flux

Let $\mathbf{p} = (x, y)$ be the coordinates of a pixel in a 2D RGB image. We represent the flux vector field $\mathbf{F}(\mathbf{p}) = \mathbf{F}(x, y) = (F_x, F_y)$ as a two-channel map with continuous values F_x, F_y, corresponding to the x and y coordinates of the flux vector, respectively. An intuitive visualization is shown in Fig. 3. In most related approaches, skeleton detection is framed as a binary classification task, for which the ground truth is a 1-pixel wide binary skeleton map. In our case, we are dealing with a *regression* problem, so we must modify the ground truth appropriately.

We divide a binary skeleton map into three non-overlapping regions: (1) *skeleton spatial context*, R_c, which is a set of pixels in the vicinity of the skeleton; (2) *skeleton pixels*, denoted by R_s; and (3) *background pixels*, R_b. In practice, we obtain R_c by dilating the binary skeleton map with a disk of radius r, and subtracting skeleton pixels R_s. Then, for each context pixel $\mathbf{p} \in R_c$, we use an efficient distance transform algorithm [18] to find its nearest skeleton pixel $p_n \in R_s$, in terms of L_2 distance. We then define the flux on the context pixel $\mathbf{p}$ as the unit direction vector that points away from $\mathbf{p}$ to p_n.[1] For the remaining pixels composed of R_s and R_b, we set the flux to $(0, 0)$. Formally, we have:

$$\mathbf{F}(\mathbf{p}) = \begin{cases} \overrightarrow{\mathbf{pp}_n} / \left|\overrightarrow{\mathbf{pp}_n}\right|, & \mathbf{p} \in R_c \\ (0, 0), & \mathbf{p} \in R_s \cup R_b, \end{cases} \tag{1}$$

where $\left|\overrightarrow{\mathbf{pp}_n}\right|$ denotes the length of the vector from pixel $\mathbf{p}$ to $\mathbf{p}_n$. We note that $\mathbf{F}(\mathbf{p})$ is defined as a unit vector field only at the context pixels in our groundtruth; at test time, the predicted field is *not* normalized.

As a representation of the spatial context associated with each skeletal pixel, our proposed spatial context flux possesses a few distinct advantages when used to detect object skeletons in the wild. Unlike most learning approaches that predict skeleton probabilities individually for each pixel, our DeepFlux method leverages consistency between flux predictions within a neighborhood around each candidate pixel. Conversely, if the true skeleton location changes, the surrounding flux field will also change noticeably. A beneficial side-effect is that our method does not rely directly on the coarse responses produced by deeper CNN layers for localizing skeletons at larger scales, which further reduces localization errors. As we show in our experiments, these properties make our method more robust to the localization of skeleton points, especially around ligature regions, and less prone to gaps, discontinuities, and irregularities caused by local mispredictions. In Sect. 3.5, we explain how we can easily and accurately recover a binary object skeleton using the magnitude and direction of the predicted flux.

[1] In fact, in the context of skeletonization of binary objects [53], this flux vector would be in the direction opposite to that of the spoke vector from a skeletal pixel to its associated boundary pixel.

3.3 Network Architecture

The network for learning the spatial context flux of skeletons closely follows the fully convolutional architecture of [35], and is shown in Fig. 4. It consists of four modules: (1) a backbone network used to extract 3D feature maps; (2) an "atrous" spatial pyramid pooling (ASPP) module [7] to enlarge the receptive field while avoiding excessive down-sampling; (3) a multi-stage feature fusion module; and (4) a flux regression and skeleton classification by convolution and up-sampling module; (5) an optional skeleton scale prediction branch, that helps to bridge the gap between skeleton extraction and a complete medial axis transform [4].

To ensure a fair comparison with previous work, we also adopt VGG16 [56] as the backbone network. As in [65], we discard the last pooling layer and the fully connected layers that follow. In the rest of the text, we call this variant DeepFlux-VGG16. The use of the atrous module is motivated by the need for a wide receptive field: when extracting skeletons we have to guarantee that the receptive field of the network is wider than the largest medial radius of an object part in the input image. The receptive field of the VGG16 backbone is 196, which is not wide enough for large objects. Furthermore, it has been demonstrated in [36] that the effective receptive field only takes up a fraction of the full theoretical receptive field. Thus, we employ ASPP to capture multi-scale information. Specifically, four parallel atrous convolutional layers with 3×3 kernels but different atrous rates (2, 4, 8, 16) are added to the last layer of the backbone, followed by a concatenation along the channel dimension. In this way, we obtain feature maps with a theoretical receptive field size of 708, which we have found to be large enough for the images we have experimented on.

To construct a multi-scale representation of the input image, we fuse the feature maps from side outputs at conv3, conv4, conv5, and ASPP layers, after convolving them with a 1×1 kernel. Since feature maps at different levels have different spatial resolutions, we resize them all to the dimensions of conv3 before concatenating them. We perform prediction on the learned flux field, after up-sampling it to the dimensions of the input image using bilinear interpolation. This is a 2-channel response map, corresponding to flux predictions $\hat{\mathbf{F}}(\mathbf{p})$ for every pixel $\mathbf{p}$ in the image.

We propose two different ways of extracting skeletons from this 2-channel response map. The first one is a simple post-processing scheme, described in Sect. 3.5. The second involves extending our network by plugging in three 3×3 convolutional layers (with 64-channel output for the first two layers), following the (up-sampled) flux field prediction

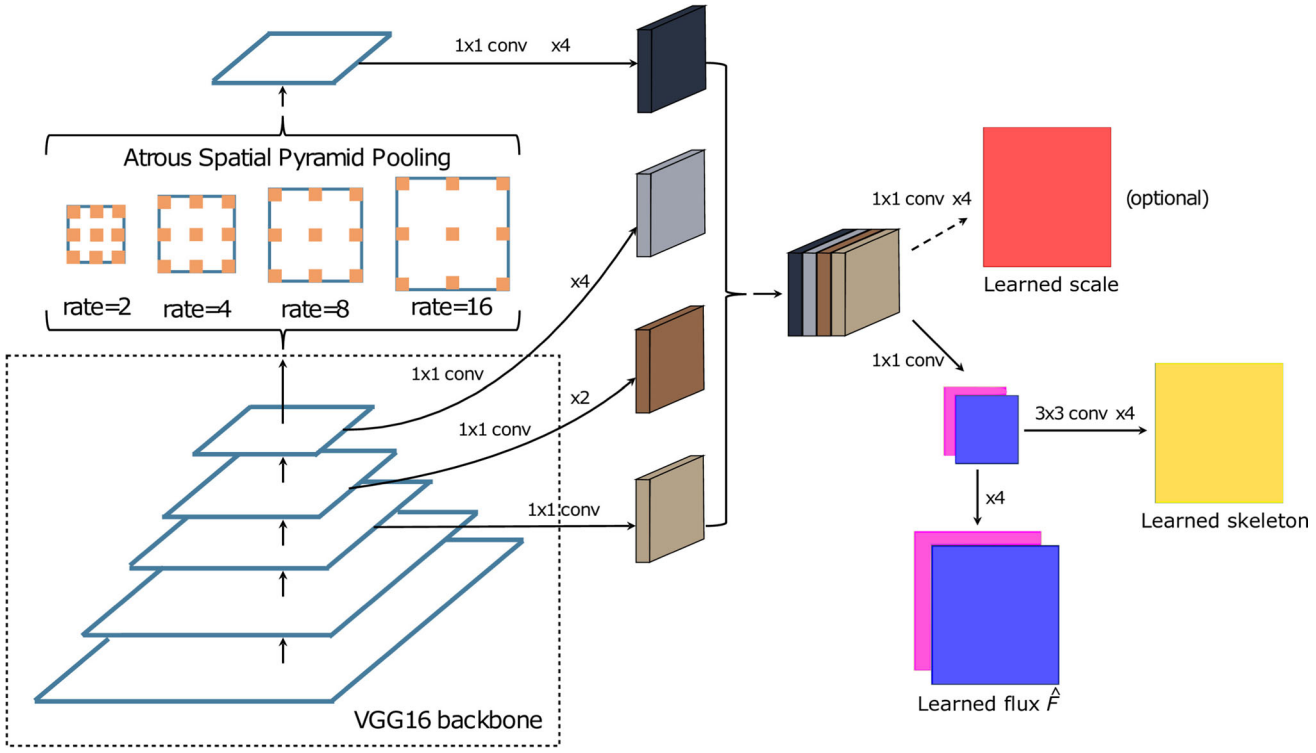


Fig. 4 End-to-end network architecture. We adopt the pre-trained VGG16 [56] (or ResNet101 [20]) with the ASPP module [7] as the backbone network. We then obtain multi-level features by concatenating features extracted from $stage3$ (or $stage2$ for a ResNet101 backbone) to $stage5$ and the ASPP layer. The network is trained to regress the spatial context flux $F = (\mathbf{F_x}, \mathbf{F_y})$ and predict a skeleton confidence score map. Our architecture can be easily augmented with a scale prediction branch to facilitate high-level tasks. In Sect. 5 we describe how to use such a variant to detect objects in aerial images

layer, which output a pixel-wise skeleton confidence score. This score can subsequently be thresholded, to produce a binary skeleton. Our network outputs both types of prediction (learned flux and skeleton confidence), as shown in Fig. 4.

We also consider an alternative architectural choice for the proposed model by replacing the VGG16 backbone with ResNet101 [20]. In this case, similar to DeepFlux-VGG16, we fuse the feature maps from different side outputs. Specifically, we apply 1×1 convolution at conv2 (whose spatial size is already 1/4 of the original image), conv3, conv4, conv5, and the ASPP layers. We then concatenate the resized side outputs together as a multi-scale representation of the input image. The following layers are kept the same as the DeepFlux-VGG16 variant. In the rest of the text, we call this variant DeepFlux-ResNet101. When not specified, we assume a DeepFlux-VGG16 architecture.

Finally, similar to previous work [49], we explore the advantages of simultaneously predicting skeleton position and scale, bridging the gap between skeleton extraction and a complete medial axis transform [4]. This also provides us with a richer representation that can find practical use in downstream tasks. To this end, we optionally include an additional branch to the DeepFlux backbone, that predicts the scale s associated with each medial point, as shown in Fig. 4.

3.4 Training Objective

We split our loss function into two terms, one for each type of output. For the *flux field* branch, we choose the L_2 loss function as our training objective. Due to a severe imbalance in the number of context and background pixels, we adopt a class-balancing strategy similar to the one in [65]. Our balanced flux loss function is

$$L_f = \sum_{\mathbf{p}\in\Omega} w_f(\mathbf{p}) \cdot \left\| \mathbf{F}(\mathbf{p}) - \hat{\mathbf{F}}(\mathbf{p}) \right\|_2^2, \tag{2}$$

where Ω is the image domain, $\hat{\mathbf{F}}(\mathbf{p})$ is the predicted flux, and $w_f(\mathbf{p})$ denotes the weight coefficient of pixel $\mathbf{p}$. The weight $w_f(\mathbf{p})$ is calculated as follows:

$$w_f(\mathbf{p}) = \begin{cases} \frac{|R_b|}{|R_c|+|R_b|+|R_s|}, & \mathbf{p} \in R_c \cup R_s \\ \frac{|R_c|+|R_s|}{|R_c|+|R_b|+|R_s|}, & \mathbf{p} \in R_b, \end{cases} \tag{3}$$

where $|R_c|$, $|R_b|$ and $|R_s|$ denote the number of context, background, and skeleton pixels, respectively.

The second branch, which predicts skeleton probability scores from the predicted flux, corresponds to a standard

Algorithm 1: Post-processing algorithm for recovering symmetry from the learned context flux $\hat{\mathbf{F}}$. $|\hat{\mathbf{F}}|$ and $\angle\hat{\mathbf{F}}$ denote the magnitude and direction of flux, respectively, and $\mathcal{N}_{\angle\hat{\mathbf{F}}(\mathbf{p})}(\mathbf{p})$ stands for the neighbor of $\mathbf{p}$ at direction $\angle\hat{\mathbf{F}}(\mathbf{p})$. λ_m is the hyper-parameter for thresholding the magnitude $|\hat{\mathbf{F}}|$. ε and δ are morphological erosion and dilation, respectively.

Input: Predicted context flux $\hat{\mathbf{F}}$, threshold λ_m
Output: Binary skeleton map M

```
function Post_Processing(F̂, λ_m)
    // initialization
    M ← False
    // find quench points near symmetry
    foreach p ∈ Ω do
        if |F̂(p)| > λ_m and |F̂(N_{∠F̂(p)}(p))| ≤ λ_m then
            M(p) ← True
    // apply morphological closing
    M ← ε_{k2}(δ_{k1}(M))
    return M
```

binary classification problem. We follow [65] and use a class-balanced cross-entropy loss function

$$L_s = -\beta \sum_{\mathbf{p}\in R_s} \log S(\mathbf{p}) - (1-\beta) \sum_{\mathbf{p}\in R_b \cup R_c} \log(1 - S(\mathbf{p})), \tag{4}$$

where S denotes the skeleton probability scores (obtained using sigmoid function), and $\beta = \frac{|R_c|+|R_b|}{|R_s|+|R_c|+|R_b|}$ is the balancing factor.

The final training objective is given by summing the two loss terms:

$$L = L_f + \lambda_1 L_s, \tag{5}$$

where λ_1 is a hyper-parameter. We set $\lambda_1 = 1$ in all our experiments.

For the optional extra scale prediction branch, we use a smoothed-L1 loss for scale regression:

$$L_{scale} = smooth_{L1}\left(\frac{\hat{s} - s}{s}\right), \tag{6}$$

where $\hat{s}$ is the predicted scale and s is the ground truth. When we also predict the skeleton scale, the overall loss is given by $L + \lambda_2 L_{scale}$, where λ_2 is a hyper-parameter that is set to 1 in all our experiments.

3.5 From Flux to Skeleton Predictions

We propose a simple post-processing procedure to recover the object skeleton from the predicted spatial context flux. As described in Eq. (1), pixels around the skeleton are labeled with unit two-dimensional vectors while the others are set to (0, 0). Thus, thresholding the magnitude of the vector field reveals the context pixels while computing the flux direction reveals the location of context pixels relative to the skeleton. We refer the reader to Fig. 2 for a visualization of the post-processing steps, listed in Algorithm 1.

Let $|\hat{\mathbf{F}}|$ and $\angle\hat{\mathbf{F}}$ be the magnitude and direction of the predicted context flux $\hat{\mathbf{F}}$, respectively. For a given pixel $\mathbf{p}$, $\angle\hat{\mathbf{F}}(\mathbf{p})$ is binned into one of 8 directions, pointing to one of the 8 neighbors, denoted by $\mathcal{N}_{\angle\hat{\mathbf{F}}(\mathbf{p})}(\mathbf{p})$. Having computed these two quantities, extracting the skeleton is straightforward: pixels close to the real object skeleton should have a high inward flux, due to a singularity in the vector field $\hat{\mathbf{F}}$, as analyzed in [12]. These pixels are defined as "quench points". Finally, we apply a morphological dilation with a disk structuring element of radius k_1, followed by a morphological erosion with a disk of radius k_2, to group quench points together and produce the object skeleton. We call this variant DeepFlux-P.

One can also *learn* to predict skeleton confidence from the predicted flux field. More precisely, as described in Sect. 3.3, we add three 3×3 convolution layers after the flux prediction layer, and train this branch in the standard manner for a binary skeleton classification problem, using a cross-entropy loss. We call this end-to-end trainable variant DeepFlux-E, and use it as our default, unless explicitly stated.

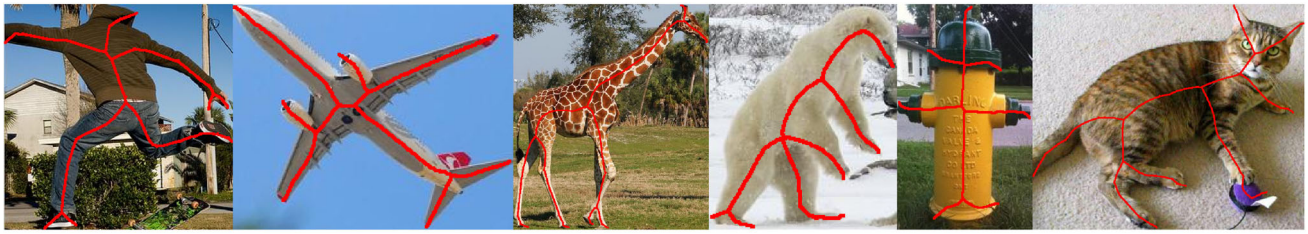
4 Experiments on Skeleton and Centerline Detection

We conduct experiments on six challenging datasets, five of which are publicly available: *SK-LARGE* [49], *SK506* [50], *WH-SYMMAX* [48], *SYM-PASCAL* [25], *SYMMAX300* [60]; and *SK-AID*, a bridge/road centerline dataset we collected ourselves from AID [63], which will also be publicly available. Some sample images are shown in Fig. 5. We note that for some of these datasets, only the skeletons of foreground objects are annotated, whereas others come with skeleton or centerline annotations for both foreground objects and background structures.

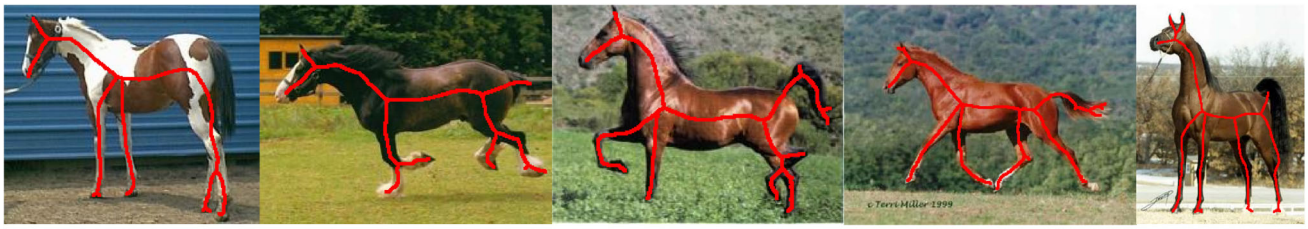
We describe the above datasets and the evaluation protocol in detail, in Sect. 4.1. We follow with implementation details in Sect. 4.2. Qualitative and quantitative results are shown in Sect. 4.3. We carry out a runtime analysis and an ablation study in Sects. 4.4 and 4.5, respectively.

4.1 Dataset and Evaluation Protocol

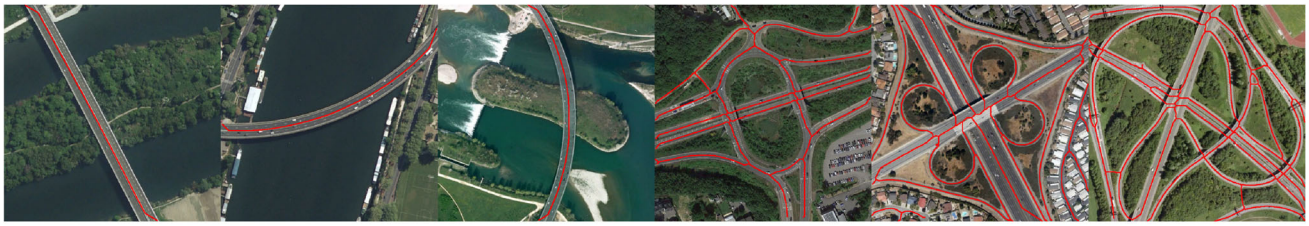
SK-LARGE [49] is a benchmark for foreground object skeleton extraction, consisting of 746 training and 745 test images. Each image in SK-LARGE is obtained by cropping an image from MS-COCO [8] so that it contains a single, cen-



(a) Samples from SK-LARGE

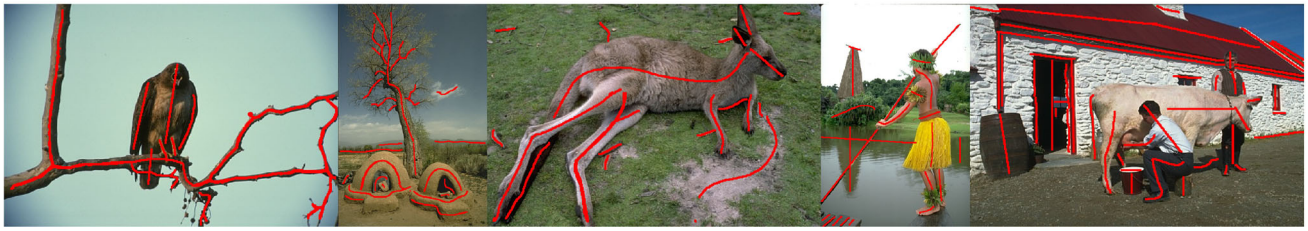


(b) Samples from WH-SYMMAX



(c) Samples from SK-AID

(d) Samples from SYM-PASCAL



(e) Samples from SYMMAX300

Fig. 5 Example images selected from different datasets, and their corresponding annotations. The ground truth annotations are thickened and drawn in red for improved visibility. Best viewed in color (Color figure online)

tered object. SK-LARGE contains various object categories including person, horse, giraffe, and man-made objects such as plane and hydrant. In this dataset, both the location and scale (the radius of the corresponding maximal disk) of each skeletal point are annotated.

SK506 [50] also referred to as SK-SMALL, is an earlier version of SK-LARGE released by the same authors [49]. There are 300 training images and 206 test images. Note that this dataset contains less training data, which might make the training of deep neural networks more challenging.

WH-SYMMAX [48] contains 328 cropped images from the Weizmann Horse dataset [5], and their skeleton point and scale annotations. The dataset is split into 228 training images and 100 test images.

SK-AID is built on AID [63], a dataset for aerial scene classification with 20 scene categories. We use 60 images for training and 40 images for testing; and focus on two object categories: bridge and road. We manually annotate the segmentation masks of roads and bridges, and then adopt a binary skeletonization algorithm [47] to obtain their centerlines as the skeleton ground truth. As shown in Fig. 5, SK-AID exhibits a large variation in skeleton orientation and curvature, as well as challenging cases of junctions of multiple skeleton branches.

SYM-PASCAL [25] is derived from the PASCAL-VOC-2011 segmentation dataset [16] for symmetry detection in the wild, and contains 648 training and 787 test images. Compared to SK-LARGE and SK506, the images from this dataset possess more complex backgrounds and variations of object appearance, including occlusions and missing parts, making it quite challenging.

SYMMAX300 [60] is built on the Berkeley Segmentation Dataset (BSDS300) [40], which contains 200 training images and 100 test images. Unlike the three datasets described above, both foreground and background regions are considered. It is noteworthy that each image in SYMMAX300 is accompanied by 5-7 symmetry annotations, corresponding to the multiple segmentation annotations existent in the BSDS300. The final local symmetry annotation is obtained by merging all available annotations for a given image, through a binary union operation.

Evaluation protocol Following previous work [49,50,60], we use precision-recall (PR) curves and the F-measure metric to evaluate skeleton detection performance in our experiments. For methods that output a skeleton probability map (including our end-to-end variant DeepFlux-E), we first apply a standard non-maximal suppression (NMS) algorithm [14]. We then threshold the thinned skeleton into a binary map and match it with the ground truth using a bi-partite matching routine that allows for small localization errors [41]. We select threshold values that yield the highest F-measure for each method-dataset combination.

For the variant DeepFlux-P, which does not directly output skeleton probabilities, we use the inverse magnitude of predicted context flux on the recovered skeleton as a surrogate for a "skeleton confidence". Thresholding at different values gives rise to a PR curve and the optimal threshold for each dataset is selected as the one producing the highest F-measure according to the formula $F = 2PR/(P + R)$. The F-measure is commonly reported as a single scalar performance index.

4.2 Implementation Details

Our implementation involves one major hyperparameter: the width of the skeleton context neighborhood r, which is set to 7 for all experiments. For the DeepFlux-P variant there are three extra hyperparameters (provided values are the ones used in our experiments): the threshold used to recover skeletal points from the predicted flux field, $\lambda_m = 0.4$ and the sizes of the structuring elements involved in the morphological operations for skeleton recovery, $k_1 = 3$ and $k_2 = 4$.

For training, we adopt standard data augmentation strategies [49,50,72]. Specifically, we resize training images to 3 different scales (0.8, 1, 1.2) and then rotate them to 4 angles (0°, 90°, 180°, 270°). We also flip them with respect to different axes (up-down, left-right, no flip). We consider two different initializations for the proposed network, one with the VGG16 [56] and one with the ResNet101 [20] model, pre-trained on ImageNet [10] and optimized using ADAM [26]. For the first 80k iterations, the learning rate is set to 10^{-5} for the backbone (VGG16 or ResNet101) layers and to 10^{-4} for the rest of the layers in the network, then reduced to 10^{-6} and 10^{-5} for the remaining 40k iterations, respectively.

We use the Caffe [23] framework to train DeepFlux. All experiments are carried out on a workstation with an Intel Xeon 16-core CPU (3.5GHz), 64GB RAM, and a single Titan Xp GPU. Training on SK-LARGE with batch size set to 1 takes about 2 hours.

4.3 Results

Comparison with other methods. We start by showing a qualitative comparison of DeepFlux-VGG16 with other skeleton detection methods, on images from WH-SYMMAX and SYM-PASCAL. As illustrated in Figs. 1 and 6, DeepFlux accurately localizes skeleton points while preserving good connectivity at junctions.

In Fig. 7 we plot the PR-curves for SK-LARGE, SK506, WH-SYMMAX, SK-AID, and SYM-PASCAL. DeepFlux significantly outperforms other methods in all cases, excelling in the high-precision regime. This is indicative of the role of local context towards more robust and accurate localization of skeleton points.

Table 1 lists the optimal F-measure score for all methods. DeepFlux-VGG16 consistently outperforms all other approaches. Specifically, DeepFlux-VGG16-E surpasses the most recent method Hi-Fi [72] by 1.2%, 2.3%, 5.0%, 6.8%, and 11.6% on SK-LARGE, SK506, WH-SYMMAX, SK-AID, and SYM-PASCAL, respectively, despite the fact that Hi-Fi uses stronger supervision during training (skeleton position *and* scale). DeepFlux-VGG16-E also outperforms LSN [31], another recent method, by 6.8%, 7.1%, 5.8%, 14.5%, and 5.1% on SK-LARGE, SK506, WH-SYMMAX, SYM-PASCAL, and SYMMAX300, respectively. It is note-

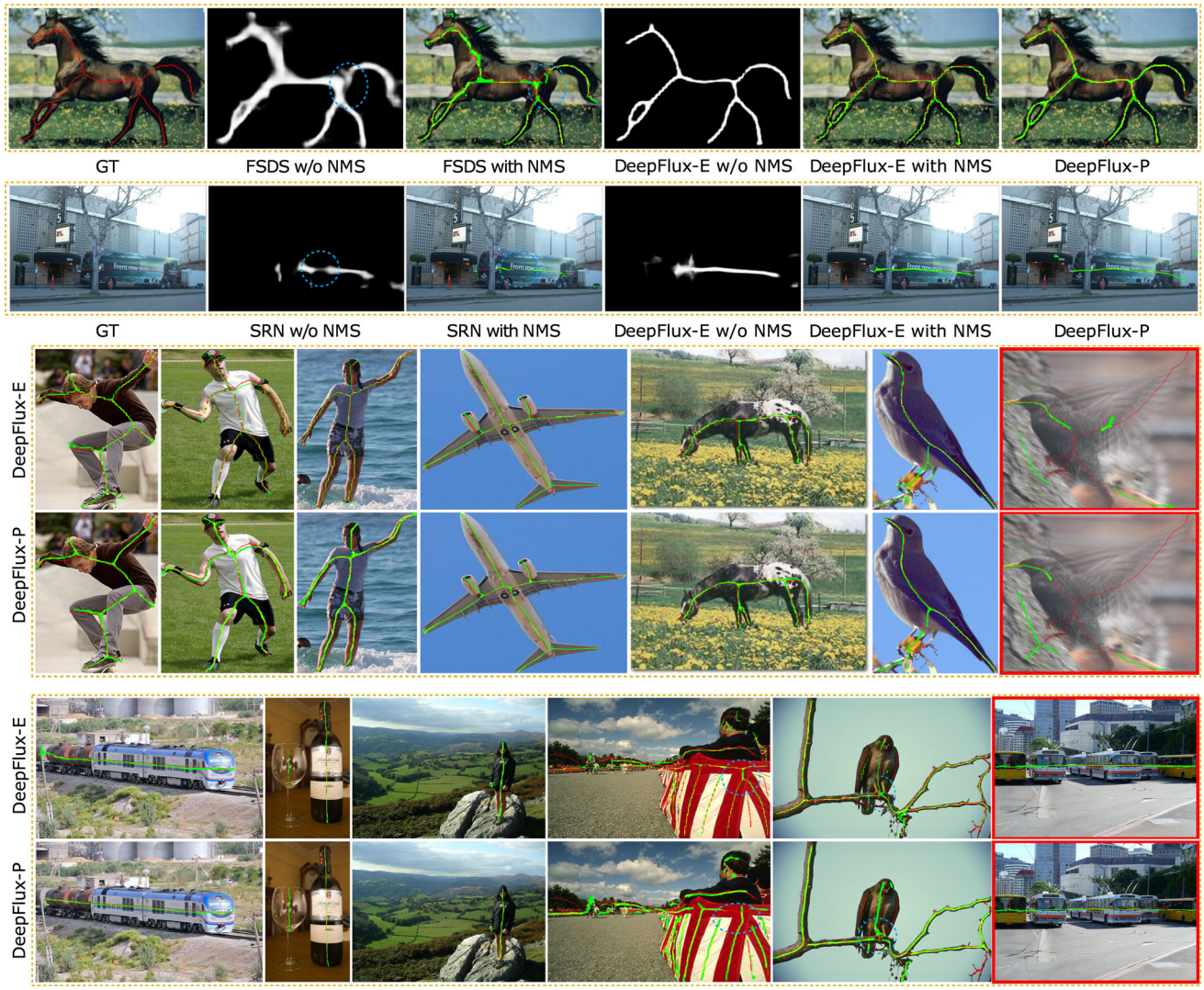


Fig. 6 Some qualitative results on SK-LARGE, WH-SYMMAX, SYM-PASCAL, and SYMMAX300. Red: GT; Green: detected skeleton; Yellow: detected skeleton and GT overlap. Qualitatively, DeepFlux-P performs similarly to the variant DeepFlux-E. Two examples of partial failure are also shown on the bottom right (enclosed by red boxes). Here DeepFlux fails to detect the skeleton on the body of the bird due to image blurring in one case. For the other case, DeepFlux detects a horizontal symmetry axis instead of a vertical one which is annotated in the ground truth (Color figure online)

worthy that the proposed DeepFlux improves over the previous state-of-the-art by more than 11% in terms of F-measure on SYM-PASCAL, whose images have more complex backgrounds and variations in object appearance. This implies that DeepFlux is better able to handle skeleton detection in complex images. For a fair comparison with previous methods, we also report results for DeepFlux, using the vanilla VGG16 architecture without the ASPP module. Barring SK-LARGE, where the proposed DeepFlux performs slightly worse than Hi-Fi [72], DeepFlux significantly outperforms competing methods on all other datasets. It is also noteworthy that Hi-Fi [72] relies on additional scale supervision during training, which is not the case for DeepFlux.

GeoSkeletonNet [66] is trained using "resolution normalization": the authors resize the images and their associated ground-truth from a size of $H \times W$ to $\sqrt{KH/W} \times \sqrt{KW/H}$ ($K = 180000$ for SYM-PASCAL and $K = 60000$ for the other datasets) before applying data augmentation. This procedure normalizes the number of pixels to a fixed value K, while keeping the aspect ratio of the images the same, factoring out the variance of resolutions across different datasets. Using the same resolution normalization protocol, DeepFlux-VGG16-E achieves a 0.758 (+0.1%), 0.730 (+0.3%), 0.863 (+1.4%), and 0.569 (+4.9%) F-score on SK-LARGE, SK506, WH-SYMMAX, and SYM-PASCAL, respectively.

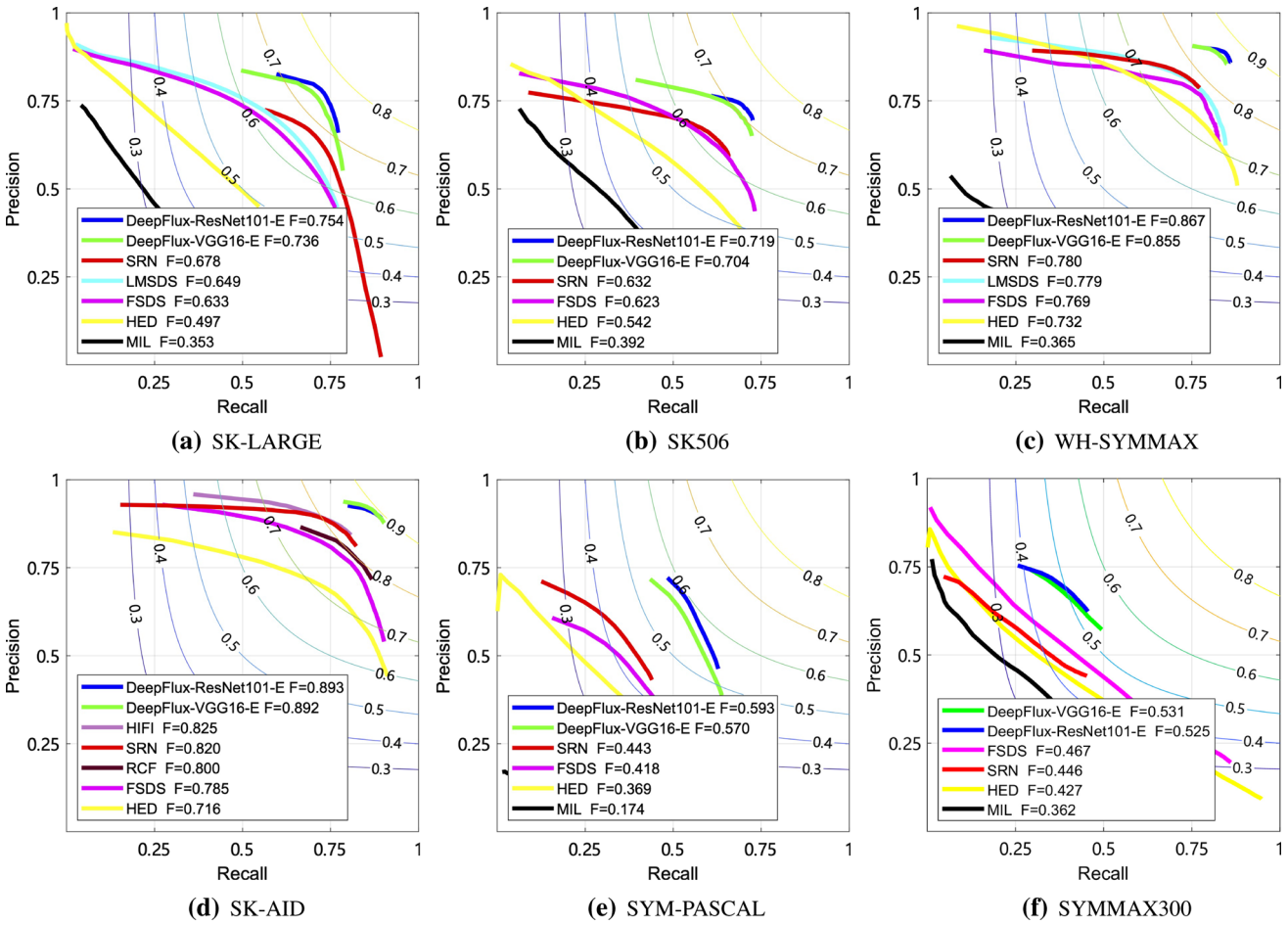


Fig. 7 Quantitative evaluation in terms of PR curves on six skeleton detection datasets. Both DeepFlux-VGG16 (in green) and DeepFlux-ResNet101 (in blue) offer high precision, especially in the high-recall regime. A stronger backbone (*e.g.*, ResNet101) leads to more accurate skeleton detection (Color figure online)

Comparison of different network backbones Using a more powerful backbone further boosts performance. DeepFlux-ResNet101-E improves over DeepFlux-VGG16-E by 1.8%, 1.5%, 1.2%, 0.1%, and 2.3% on SK-LARGE, SK506, WH-SYMMAX, SK-AID, and SYM-PASCAL, respectively. The modest gains from the more powerful ResNet on SK-AID can potentially be attributed to the significantly lower variation of skeleton scales in that dataset; the capacity of VGG16 seems to be sufficient to already achieve close to 90% accuracy. Curiously, DeepFlux-ResNet101 performs slightly worse than DeepFlux-VGG16 on SYMMAX300. Our hypothesis is that, because of the multiple—potentially conflicting annotations—per image in this dataset, the lower capacity of the VGG16 may act as a regularizer, leading to slightly better performance.

Post-processing versus end-to-end training DeepFlux-E (end-to-end) performs slightly better than DeepFlux-P (post-processing), in all cases. In particular, DeepFlux-VGG16-E outperforms DeepFlux-VGG16-P by 1.2% and 0.6% on SYM-PASCAL and SYMMAX300, respectively. As shown in Fig. 6 for qualitative results on SYMMAX300 (see the blue dashed circles), DeepFlux-E preserves better the connectivity at ligature areas than DeepFlux-P, which may only have a few quench points instead of a set of connected ones due to direction discretization into 8 bins. DeepFlux-E also enjoys a slightly faster runtime, as shown in the comparison in Table 2.

Failure cases Despite the effectiveness of DeepFlux inaccurately detecting object skeletons in images, there are some challenging cases where the model fails partially. An example is illustrated on the middle right of Fig. 6, where the skeleton of the body of the bird is not detected due to severe image blurring. Another example of failure is shown on the bottom right in Fig. 6, where DeepFlux fails to capture the symmetry of each bus instance individually, detecting instead the horizontal symmetry axis of the entire cluster.

Table 1 Quantitative comparison in terms of F-measure

Methods	Backbone	SK-LARGE	SK506	WH-SYMMAX	SK-AID	SYM-PASCAL	SYMMAX300
MIL [60]	VGG16'	0.353	0.392	0.365	–	0.174	0.362
HED [65]	VGG16'	0.497	0.541	0.732	0.790	0.369	0.427
RCF [34]	VGG16'	0.626	0.613	0.751	0.800	0.392	–
FSDS* [50]	VGG16'	0.633	0.623	0.769	–	0.418	0.467
LMSDS* [49]	VGG16'	0.649	0.621	0.779	–	–	–
SRN [25]	VGG16'	0.678	0.632	0.780	0.820	0.443	0.446
LSN [31]	VGG16'	0.668	0.633	0.797	–	0.425	0.480
Hi-Fi* [72]	VGG16	**0.724**	0.681	0.805	0.824	0.454	–
DeepFlux-P'	VGG16'	0.714	0.687	0.845	0.863	0.492	0.486
DeepFlux-E'	VGG16'	0.715	**0.688**	**0.847**	**0.871**	**0.508**	**0.519**
DeepFlux-P	VGG16	0.734	0.703	0.850	0.878	0.558	0.525
DeepFlux-E	VGG16	**0.736**	**0.704**	**0.855**	**0.892**	**0.570**	**0.531**
DeepFlux-P	ResNet101	0.750	0.717	0.861	0.883	0.585	0.517
DeepFlux-E	ResNet101	**0.754**	**0.719**	**0.867**	**0.893**	**0.593**	**0.525**

The best results of each corresponding group are marked in bold
*Indicates scale supervision was also used. Results for competing methods are from the corresponding papers for all datasets except the self-collected SK-AID, on which the results are obtained using the corresponding open-source implementation. DeepFlux-E performs slightly better than DeepFlux-P. VGG16' denotes the vanilla VGG architecture without using the ASPP module

Table 2 Runtime and performance on SK-LARGE. For DeepFlux-P, we list the total inference (GPU) + post-processing (CPU) time

Method	F-measure	Runtime (in sec)
HED [65]	0.497	0.014
FSDS [50]	0.633	0.017
LMSDS [49]	0.649	0.019
LSN [31]	0.668	0.021
SRN [25]	0.678	0.016
Hi-Fi [72]	0.724	0.030
DeepFlux-VGG16-P (ours)	0.734	0.017
DeepFlux-VGG16-E (ours)	0.736	0.014
DeepFlux-ResNet101-P (ours)	0.750	0.021
DeepFlux-ResNet101-E (ours)	0.754	0.018

4.4 Runtime Analysis

In Table 2 we compare the runtime of DeepFlux to alternatives. Since competing models typically use the VGG16 backbone, we mainly employ the DeepFlux-VGG16 variant in our analysis, to keep the comparison fair. As shown in Table 2, DeepFlux is as fast as competing methods while achieving superior performance. Inference of DeepFlux-VGG16-E on the GPU takes on average 14 ms for a 300×200 image, which is faster than other methods. The DeepFlux-VGG16-P variant requires on average an extra 3 ms on the CPU, for post-processing.

4.5 Ablation Study

We study the contribution of the two main modules (ASPP module and flux representation) to skeleton detection on SK-LARGE and SYM-PASCAL, by removing them one at a time from the VGG16 backbone. We conduct four experiments corresponding to the four possible combinations of each module being present or not.

When the spatial context flux representation is not used, we train the model with the same architecture, but for binary classification using binary cross-entropy loss. The baseline model is trained without ASPP and spatial context flux representation. As depicted in Table 3, the ASPP module that offers a larger receptive field, results in an improvement of 1.9% on SK-LARGE and 5.1% on SYM-PASCAL, compared to the baseline model. This confirms that a large receptive field is beneficial for skeleton/symmetry extraction. We then remove the ASPP module and train the model using the proposed flux representation, which yields an improvement of 0.8% on SK-LARGE and 3.5% on SYM-PASCAL. These gains are complementary to each other; indeed, combining both the ASPP and the flux representation, improves performance over the baseline by 2.9% on SK-LARGE and 9.7% on SYM-PASCAL.

We also study the effect of the size r of the neighborhood within which context flux is computed. We conduct experiments with different radii, ranging from $r = 3$ to $r = 11$, with a step of 2, on the SK-LARGE and SYM-PASCAL datasets. The best results are obtained for $r = 7$, and using smaller or larger values seems to slightly decrease

Table 3 Ablation study on the effect of the spatial context flux representation and the ASPP module on the performance in terms of F-measure

Dataset	Context flux	ASPP	F-measure
SK-LARGE			0.707
		✓	0.726
	✓		0.715
	✓	✓	**0.736**
SYM-PASCAL			0.473
		✓	0.524
	✓		0.508
	✓	✓	**0.570**

The best results of each corresponding group are marked in bold

Table 4 Ablation study on the influence of the context size r on the performance in terms of F-measure

Dataset	$r=3$	$r=5$	$r=7$	$r=9$	$r=11$
SK-LARGE	0.733	0.733	**0.736**	0.732	0.730
SYM-PASCAL	0.560	0.563	**0.570**	0.561	0.562

The best results of each corresponding group are marked in bold

performance. Our understanding is that a narrower spatial context neighborhood provides less contextual information to predict the final skeleton map. On the other hand, using a wider neighborhood may increase the chance for mistakes in flux prediction around areas of severe discontinuities, such as the areas around boundaries of thin objects that are fully contained in the context neighborhood. DeepFlux does not appear to be sensitive to the value of r, as shown in Table 4.

Finally, one may argue that simply using a dilated ground truth when training the network for skeleton classification is sufficient to make the model more robust in accurately localizing skeletal points. To examine if this is the case, we removed the flux module and retrained our VGG16-based model on the same dilated skeletons we used to compute the spatial context flux ground truth, using a binary cross-entropy loss instead. Without spatial context flux representation, the performance drops from $F = 0.736$ to $F = 0.697$ (-3.9%) on SK-LARGE and from $F = 0.570$ to $F = 0.490$ (-8%) on SYM-PASCAL, demonstrating the effectiveness of our proposed representation for accurate localization.

5 Application to Object Detection in Aerial Images

We consider an application of simultaneously predicting skeleton position and scale, for the task of detecting large vehicles in remote sensing imagery. For a fair comparison with other methods, we use ResNet101 as the backbone. The stride between block3 and block4 is set to 1, and all the layers in block4 are replaced with dilated convolution layers.

Table 5 Comparison with some state-of-the-art methods dedicated for object detection in remote sensing images on DOTA [64]

Methods	FPN	Large-vehicles
FR-O [64]	–	38.02
RRPN [37]	–	56.19
R2CNN [24]	–	50.91
R-DFPN [68]	✓	50.94
RoI Transformer [13]	–	62.97
Ours	–	65.56

The results for the other methods are from [13]

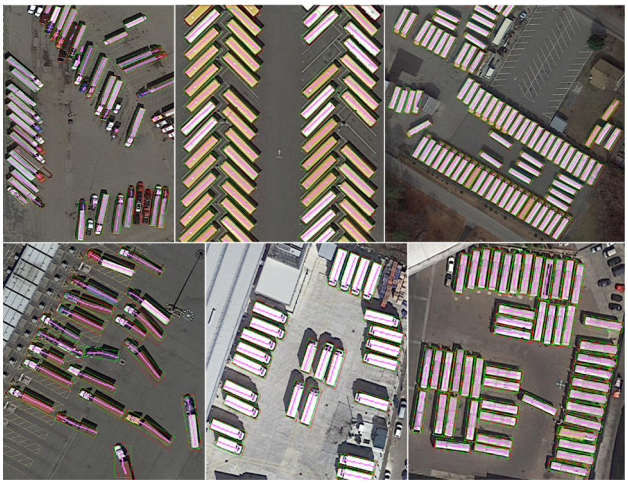


Fig. 8 Qualitative visualization of some large vehicle detection results on the DOTA dataset. Pink line: predicted skeleton segment; Green box: large vehicle detection; Red box: GT; Best viewed in the electronic version (Color figure online)

Predicting the scale associated with each skeleton pixel allows us to generate the object mask in a straightforward way. Let $\hat{s}_i$ denote the predicted scale for a skeleton pixel x_i in a skeleton segment (*i.e.*, a connected component of the binary object skeleton). We obtain the object mask as $O = \cup_{i=1}^{N} D_i$, where N is the number of the skeleton pixels in the segment, and D_i is the disk of radius $\hat{s}_i$ centered at x_i. We use the bounding box of O as our final detection, and the mean magnitude value of the enclosed spatial context flux as a proxy for the classification score for evaluation purposes. As depicted in Table 5, we achieve competitive performance against state-of-the-art methods dedicated for object detection in aerial images. Qualitative detection results are shown in Fig. 8.

6 Conclusions

We have proposed DeepFlux, a novel approach for accurate skeleton detection in the wild. In contrast to classical learning-based methods that consider skeleton detection as a binary classification problem, we learn to regress a 2D vector field of "context flux". Context flux is a reliable intermediate cue for skeleton point localization, either through simple post-processing or end-to-end training. The proposed approach alleviates many limitations (*e.g.*, poor localization) of previous methods, and performs very well in handling ligature points, and skeletons of objects at large spatial scales, while also being very fast ($\sim 14-17ms$ for detection on a Titan Xp GPU). Our experiments on six challenging benchmarks demonstrate that DeepFlux consistently improves over the state-of-the-art both quantitatively and qualitatively.

While the skeleton represents a powerful shape representation in support of many tasks, it lacks the dual boundary/region encoding offered by the medial axis transform (MAT), since skeleton points do not encode the scale of the maximal inscribed disk. We have extended our framework to explicitly recover both skeleton position and scale, significantly enhancing the representation power and utility of our skeletons, as demonstrated on an object detection/segmentation task of vehicle detection in remote sensing imagery.

Acknowledgements This work was supported in part by NSFC 61936003 and 61703171, and the Major Project for New Generation of AI under Grant No. 2018AAA0100400. Yongchao Xu was supported by the Young Elite Scientists Sponsorship Program by CAST. The work of Xiang Bai was supported by the National Program for Support of Top-Notch Young Professionals and in part by the Program for HUST Academic Frontier Youth Team. Sven Dickinson and Kaleem Siddiqi would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for research funding.

References

1. Ahn, J., Cho, S., & Kwak, S. (2019). Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 2209–2218).
2. Bai, M., & Urtasun, R. (2017). Deep watershed transform for instance segmentation. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 2858–2866).
3. Bai, X., Wang, X., Latecki, L. J., Liu, W., & Tu, Z. (2009). Active skeleton for non-rigid object detection. In *Proceedings of IEEE international conference on computer vision* (pp. 575–582).
4. Blum, H. (1973). Biological shape and visual science (part i). *Journal of Theoretical Biology*, *38*(2), 205–287.
5. Borenstein, E., & Ullman, S. (2002). Class-specific, top-down segmentation. In *Proceedings of European conference on computer vision* (pp. 109–122).
6. Chen, L. C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., & Adam, H. (2018). Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 4013–4022).
7. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848.
8. Chen, X., Fang, H., Lin, T. Y., Vedantam, R., Gupta, S., Dollár, P., & Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. CoRR **abs/1504.00325**.
9. Ci, H., Wang, C., & Wang, Y. (2018). Video object segmentation by learning location-sensitive embeddings. In *Proceedings of European conference on computer vision* (pp. 501–516).
10. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Li, F. F. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 248–255).
11. Dickinson, S. J. (2009). *Object categorization: Computer and human vision perspectives*. Cambridge: Cambridge University Press.
12. Dimitrov, P., Damon, J. N., & Siddiqi, K. (2013). Flux invariants for shape. In *Proceedings of IEEE international conference on computer vision and pattern recognition*.
13. Ding, J., Xue, N., Long, Y., Xia, G. S., & Lu, Q. (2019). Learning RoI transformer for oriented object detection in aerial images. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 2849–2858).
14. Dollár, P., & Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(8), 1558–1570.
15. Dufresne-Camaro, C. O., Rezanejad, M., Tsogkas, S., Siddiqi, K., & Dickinson, S. (2020). Appearance shock grammar for fast medial axis extraction from real images. In *Proceedings of IEEE international conference on computer vision and pattern recognition*.
16. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.
17. Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55–79.
18. Felzenszwalb, P. F., & Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of Computing*, *8*(1), 415–428.
19. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., & Fitzgibbon, A. (2011). Efficient regression of general-activity human poses from depth images. In *Proceedings of IEEE international conference on computer vision* (pp. 415–422).
20. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 770–778).
21. Jang, J. H., & Hong, K. S. (2001). A pseudo-distance map for the segmentation-free skeletonization of gray-scale images. In *Proceedings of IEEE international conference on computer vision* (vol. 2, pp. 18–23).
22. Jerripothula, K. R., Cai, J., Lu, J., & Yuan, J. (2017). Object co-skeletonization with co-segmentation. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 3881–3889).
23. 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 *Proceedings of ACM multimedia* (pp. 675–678).
24. Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., Fu, P., & Luo, Z. (2017). R2CNN: Rotational region CNN for orientation robust scene text detection. Preprint arXiv:1706.09579.

25. Ke, W., Chen, J., Jiao, J., Zhao, G., & Ye, Q. (2017) SRN: Side-output residual network for object symmetry detection in the wild. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 302–310).
26. Kinga, D., & Adam, J. B.: A method for stochastic optimization. In *Proceedings of international conference on learning representations* (vol. 5).
27. Kreiss, S., Bertoni, L., & Alahi, A. (2019) PifPaf: Composite fields for human pose estimation. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 11977–11986).
28. Levinshtein, A., Sminchisescu, C., & Dickinson, S. (2013). Multi-scale symmetric part detection and grouping. *International Journal of Computer Vision*, *104*(2), 117–134.
29. Lindeberg, T. (1998). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, *30*(2), 117–156.
30. Lindeberg, T. (2013). Scale selection properties of generalized scale-space interest point detectors. *Journal of Mathematical Imaging and Vision*, *46*(2), 177–210.
31. Liu, C., Ke, W., Qin, F., & Ye, Q. (2018). Linear span network for object skeleton detection. In *Proceedings of European conference on computer vision* (pp. 136–151).
32. Liu, T. L., Geiger, D., & Yuille, A. L. (1998). Segmenting by seeking the symmetry axis. In *Proceedings of international conference on pattern recognition* (vol. 2, pp. 994–998).
33. Liu, X., Lyu, P., Bai, X., & Cheng, M. M. (2017). Fusing image and segmentation cues for skeleton extraction in the wild. In *Proceedings of ICCV workshop on detecting symmetry in the wild* (vol. 6, p. 8).
34. Liu, Y., Cheng, M. M., Hu, X., Wang, K., & Bai, X. (2017). Richer convolutional features for edge detection. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 5872–5881).
35. Long, J., Shelhamer, E., & Darrell, T. (2015) Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 3431–3440).
36. Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of advances in neural information processing systems* (pp. 4898–4906).
37. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., et al. (2018). Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, *20*(11), 3111–3122.
38. Maninis, K. K., Pont-Tuset, J., Arbeláez, P., & Van Gool, L. (2018). Convolutional oriented boundaries: From image segmentation to high-level tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 819–833.
39. Marr, D., & Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London B: Biological Sciences*, *200*(1140), 269–294.
40. 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 *Proceedings of IEEE international conference on computer vision* (vol. 2, pp. 416–423).
41. Martin, D. R., Fowlkes, C. C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(5), 530–549.
42. Máttyus, G., Luo, W., & Urtasun, R. (2017). Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE international conference on computer vision*.
43. Mattyus, G., Wang, S., Fidler, S., & Urtasun, R. (2015). Enhancing road maps by parsing aerial images around the world. In *Proceedings of the IEEE international conference on computer vision* (pp. 1689–1697).
44. Nedzved, A., Ablameyko, S., & Uchida, S. (2006). Gray-scale thinning by using a pseudo-distance map. In *Proceedings of IEEE international conference on pattern recognition*.
45. Peng, S., Liu, Y., Huang, Q., Zhou, X., & Bao, H. (2019). PVNet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 4561–4570).
46. Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. *IEEE Transactions on Multimedia*, *15*(5), 1110–1120.
47. Shen, W., Bai, X., Hu, R., Wang, H., & Latecki, L. J. (2011). Skeleton growing and pruning with bending potential ratio. *Pattern Recognition*, *44*(2), 196–209.
48. Shen, W., Bai, X., Hu, Z., & Zhang, Z. (2016). Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images. *Pattern Recognition*, *52*, 306–316.
49. Shen, W., Zhao, K., Jiang, Y., Wang, Y., Bai, X., & Yuille, A. (2017). Deepskeleton: Learning multi-task scale-associated deep side outputs for object skeleton extraction in natural images. *IEEE Transactions on Image Processing*, *26*(11), 5298–5311.
50. Shen, W., Zhao, K., Jiang, Y., Wang, Y., Zhang, Z., & Bai, X. (2016). Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 222–230).
51. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011) Real-time human pose recognition in parts from single depth images. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 1297–1304).
52. Siddiqi, K., Bouix, S., Tannenbaum, A., & Zucker, S. W. (2002). Hamilton-jacobi skeletons. *International Journal of Computer Vision*, *48*(3), 215–231.
53. Siddiqi, K., & Pizer, S. M. (2008). *Medial Representations: Mathematics*., Algorithms and Applications Berlin: Springer.
54. Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., & Zucker, S. W. (1999). Shock graphs and shape matching. *International Journal of Computer Vision*, *35*(1), 13–32.
55. Sie Ho Lee, T., Fidler, S., & Dickinson, S. (2013). Detecting curved symmetric parts using a deformable disc model. In *Proceedings of IEEE international conference on computer vision* (pp. 1753–1760).
56. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of international conference on learning representations*.
57. Sironi, A., Lepetit, V., & Fua, P. (2014). Multiscale centerline detection by learning a scale-space distance transform. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 2697–2704).
58. Trinh, N. H., & Kimia, B. B. (2011). Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *International Journal of Computer Vision*, *2*, 215–240.
59. Tsogkas, S., & Dickinson, S. (2017) AMAT: Medial axis transform for natural images. In *Proceedings of IEEE international conference on computer vision* (pp. 2727–2736).
60. Tsogkas, S., & Kokkinos, I. (2012). Learning-based symmetry detection in natural images. In *Proceedings of European conference on computer vision* (pp. 41–54).
61. Wang, Y., Xu, Y., Tsogkas, S., Bai, X., Dickinson, S., & Siddiqi, K. (2019). Deepflux for skeletons in the wild. In *Proceedings of IEEE*

international conference on computer vision and pattern recognition (pp. 5287–5296).
62. Wei, S. E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 4724–4732).
63. Xia, G., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., et al. (2017). AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions Geoscience and Remote Sensing*, *55*(7), 3965–3981.
64. Xia, G. S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., & Zhang, L. (2018) DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 3974–3983).
65. Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of IEEE international conference on computer vision* (pp. 1395–1403).
66. Xu, W., Parmar, G., & Tu, Z. (2019). Geometry-aware end-to-end skeleton detection. In *British Machine Vision Conference*.
67. Xu, Y., Wang, Y., Zhou, W., Wang, Y., Yang, Z., & Bai, X. (2019). Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, *28*(11), 5566–5579.
68. Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M., et al. (2018). Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, *10*(1), 132.
69. Yu, Z., & Bajaj, C. (2004). A segmentation-free approach for skeletonization of gray-scale images via anisotropic vector diffusion. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 415–420).
70. Zhang, Q., & Couloigner, I. (2007). Accurate centerline detection and line width estimation of thick lines using the radon transform. *IEEE Transactions on Image Processing*, *16*(2), 310–316.
71. Zhang, Z., Shen, W., Yao, C., & Bai, X. (2015). Symmetry-based text line detection in natural scenes. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 2558–2567).
72. Zhao, K., Shen, W., Gao, S., Li, D., & Cheng, M. M. (2018). Hi-fi: Hierarchical feature integration for skeleton detection. In *Proceedings of international joint conference on artificial intelligence* (pp. 1191–1197).
73. Zhu, S. C., & Yuille, A. L. (1996). Forms: A flexible object recognition and modelling system. *International Journal of Computer Vision*, *20*(3), 187–212.
74. Zucker, S. W. (2012). Local field potentials and border ownership: A conjecture about computation in visual cortex. *Journal of Physiology-Paris*, *106*, 297–315.