KPNet: Towards Minimal Face Detector

Guanglu Song,1,3 Yu Liu,2∗ Yuhang Zang,1 Xiaogang Wang,2 Biao Leng,3,4 Qingsheng Yuan5

1SenseTime X-Lab
2The Chinese University of Hong Kong, Hong Kong
3School of Computer Science and Engineering, Beihang University, Beijing 100191, China
4Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, 100191
5National Computer network Emergency Response technical Team/Coordination Center of China

Abstract

The small receptive field and capacity of minimal neural networks limit their performance when using them to be the backbone of detectors. In this work, we find that the appearance feature of a generic face is discriminative enough for a tiny and shallow neural network to verify from the background. And the essential barriers behind us are 1) the vague definition of the face bounding box and 2) tricky design of anchor-boxes or receptive field. Unlike most top-down methods for joint face detection and alignment, the proposed KPNet detects small facial keypoints instead of the whole face by in the bottom-up manner. It first predicts the facial landmarks from a low-resolution image via the well-designed fine-grained scale approximation and scale adaptive soft-argmax operator. Finally, the precise face bounding boxes, no matter how we define it, can be inferred from the keypoints. Without any complex head architecture or meticulous network designing, the KPNet achieves state-of-the-art accuracy on generic face detection and alignment benchmarks with only ∼1M parameters, which runs at 1000fps on GPU and is easy to perform real-time on most modern frontend chips.

Introduction

The performance of face detection has been constantly improved thanks to the anchor-based mechanism (Girshick 2015; Ren et al. 2015) with the top-down strategy. By simply assigning dense anchor templates in complex models, we can obtain a face detector with excellent performance. In the current state-of-the-art research, the essence of face detection is how to design the receptive field adaptive to large scale-variance. With the emergence of some seminal works (Hu and Ramanan 2017; Zhang et al. 2017d; Najibi et al. 2017) to explore the relationship between the receptive field and the large scale-variance, the performance of face detection is further improved. Inspired by this, the FPN-style framework (Lin et al. 2017) has become a priority choice for researchers and it can effectively enhance the performance of face detectors to handle faces with different scales. Encouraged by these insights, most of the state-of-the-art algorithms (Tang et al. 2018; Chi et al. 2018; Li et al. 2018) construct adaptive receptive fields to detect faces. They either detect targets with different scales at different levels of the network or detect targets by fusing enhanced features generated by different levels. With the assistance of complex deep backbone (He et al. 2016), improving face detection performance with refined receptive field (Zhang et al. 2017d; Najibi et al. 2017) or embedding new enhanced modules (Tang et al. 2018; Chi et al. 2018; Li et al. 2018) has become the guidance in the field of face detection. However, these top-down approaches with complex backbone networks lead to a heavy computational burden, even though some novel works (Song et al. 2018; Liu et al. 2017) are proposed to accelerate them. Under the constraints of these current mechanisms, we naturally raise a question: can large scale-variance be solved only through a deeper and more complex backbone with well-designed strategies?

Keep the bottleneck of current research in mind, this paper tries to seek the answer to this question. We re-explore
face detection from the two aforementioned essential factors: the receptive field and the large scale-variance. The above question is decomposed into the following more detailed sub-problems and different controlled experiments are performed on FDDB to seek answers.

- It’s a common practice to up-sample the images to 480 \times 640 or even 800 \times 1000 which facilitates better performance based on a complex backbone network with a large receptive field. Will the lightweight network fail in this configuration?
- How do complex and lightweight networks with different receptive fields perform on low-resolution input images?

We design detection backbones of various depths with different receptive fields by modifying ResNet50 (He et al. 2016). Four detectors CNN-50L, CNN-41L, CNN-26L and CNN-17L with depth 50, 41, 26 and 17 are performed on FDDB. The training set is same with (Liu et al. 2017) and the recall of top 100 proposals for each image is used for evaluation. Results are shown in Fig. 1 and the anchor setting is that \( A = \{\lceil16\sqrt{2}, 16\sqrt{2}\rceil, \lceil32\sqrt{2}, 32\sqrt{2}\rceil, \lceil64\sqrt{2}, 64\sqrt{2}\rceil\} \) for image resolution [200, 400] to detect face scale [16, 128], 2.4 for image resolution [400, 600] to detect face scale [32, 256] and 4.4 for image resolution [800, 1000] to detect face scale [64, 512]. According to the results, the former questions can be explained. When the lightweight network adopts the high-resolution image, even if the appropriate anchor templates are assigned, it still fails in this configuration due to the limitation of the receptive field. It’s worth noting that shrinking the image to low resolution with lightweight backbone can still lead to the comparable performance to the deeper and more complex backbone.

However, potential barriers still exist behind this discovery. The accurate face boxes heavily rely on the tricky design of anchor boxes or receptive field and also, the vague definition of face boxes (e.g., face boxes in FDDDB are defined by the ellipse) can easily degrade the performance. Fortunately, bottom-up methods can effectively get rid of the bottleneck in top-down mechanism via converting boxes to keypoints (Law and Deng 2018; Zhou, Zhuo, and Krähenbühl 2019). So these ideas naturally lead to a simple, lightweight but accurate framework KPNet where two essential factors are embedded into it, one is to shrink the input image to low resolution with lightweight backbone and the other is to shrink the face concept from box to keypoints to skip the deficiency of general pipeline.

Beyond the general face detection pipeline, the precise facial keypoints can be located first via the carefully designed algorithm and then the accurate face boxes can be inferred by it. So that it can perform as a bottom-up approach to joint face detection and alignment. Different from other top-down algorithms (Zhang et al. 2016; King 2009) for joint face detection and alignment, KPNet bypasses the vague definition of bounding boxes and takes advantage of the less uncertainty definition of keypoints. Moreover, it’s different from the bottom-up approaches in pose estimation where each landmark is independently predicted and associative embedding (Newell, Huang, and Deng 2017; Law and Deng 2018) is used to group them into an instance. The well-designed fine-grained scale approximation in KPNet can potentially imply the group of landmarks and with the scale adaptive soft-argmax, it can straightforwardly predict the landmarks belonging to the same face.

To summarize, the contributions of this paper are as follows:

1) According to the re-exploration on face detection and the advantages of shrinking target concept from box to keypoints, we propose a novel KPNet with the simple, lightweight but accurate mechanism for the generic face (>20 px) detection and alignment.

2) We propose the fine-grained scale approximation and scale-based customization soft-argmax operator to improve the performance by a large margin.

3) Different from all of the joint face detection and alignment methods that adopt the top-down pipeline, KPNet follows the bottom-up mechanism and the more precise definition of landmarks than boxes enables the better performance.

4) Without bells and whistles, KPNet can achieve the SOTA performance on generic face detection benchmarks FDDDB, AFW, MALF, and face alignment benchmark AFLW. And also the model inference speed with the offline application can achieve \( \sim 1000\text{fps} \) on GTX 1080Ti.

Related Works

Face detection. Since the emergence of the powerful CNN (He et al. 2016), the performance of face detection has been improved by a large margin. With the success of anchor-based methods such as Fast RCNN (Girshick 2015) and Faster RCNN (Ren et al. 2015) on object detection, several different approaches (Li et al. 2018; Yang et al. 2017) has determined its position in the field of face detection. More recently, the FPN-style framework (Lin et al. 2017) encourages the researchers to explore the relationship between the receptive field of the face detector and the anchor design skills (Zhang et al. 2017d). Benefiting from these explorations, detecting faces with different scales from different layers in a single network (Najibi et al. 2017; Tang et al. 2018; Yang et al. 2017) has determined its position in the field of face detection. Several works (Li et al. 2018; Chi et al. 2018) detect a face from the feature fusion of different layers and the enhanced feature make it robust for scale-variance. The deeper and complex backbone with the FPN-style framework achieves the new state-of-the-art and this idea of detector design has dominated face detection for many years. Unfortunately, the accurate face detection heavily relies on the tricky design of anchor boxes or receptive field. Moreover, the vague definition of face boxes makes it hard to generalize to generic face detection.

Face alignment. Face alignment refers to facial landmark detection and it mainly focuses on identifying the geometry structure of the human face. The CNN-based face alignment methods can be divided into two categories, i.e., coordinate regression model and heatmap regression model.

The coordinate regression model directly regresses the facial landmarks from the input image. Many works (Dong et al. 2018b; Feng et al. 2018) have the advantage of explicit inference of landmarks without any post-processing.
However, these regression-based methods are not performing as well as heatmap regression models. The heatmap regression models generate likelihood heatmaps for each key-point, respectively. In these heatmap-based methods, hourglass networks (Newell, Yang, and Deng 2016) becomes the backbone of many works (Newell, Yang, and Deng 2016; Deng et al. 2017) due to its capabilities of obtaining multiscale information. Some works (Lv et al. 2017; Sun, Wang, and Tang 2013) have adopted facial parts to aid face alignments tasks and LAB (Wu et al. 2018) uses more precise facial boundary to assist in detecting facial landmarks. Some works (Lv et al. 2017; Sun, Wang, and Tang 2013) due to its capabilities of obtaining multi-scale information. Some works (Lv et al. 2017; Sun, Wang, and Tang 2013) have adopted facial parts to aid face alignments tasks and LAB (Wu et al. 2018) uses more precise facial boundary to assist in detecting facial landmarks. 

Joint face detection and alignment. Face alignment and face detection are closely related, yet few works (Zhang et al. 2017) where CNN is capable of approximating the scale information in the low-resolution image, we convert the boxes regression to fine-grained face scale classification for each pixel in a feature map. Different from (Liu et al. 2017) where only the existing scales are predicted to select valid layers from the image pyramid, in KPNet, we add additional spatial information to scale approximation. The scalemap is generated by fine-grained scale approximation that consists of only one convolutional layer with kernel size 3 is used. It is a probability map $M$ with dimension $H' \times W' \times S$, where $S$ is the predefined number of scales. Given an image $I$ with face boxes $[x, y, h, w]$ where $(x, y)$ means the face center and $h, w$ represent the height and width of the face, the $M$ is firstly initialized to 0 and then we calculate the active channel index $b$ as:

$$b = 10 \times \left( \log_2 \frac{\max(h, w)}{I_{\max}} - 5 \right),$$

where $b$ is the index from 1 to $S$ and $I_{\max}$ represents the max edge of $I$. The minimum detected face size for $I_{\max}=1280$ in 720P image is 20px. In Eq. 1, the face size from $2^5$ to $2^3$ can be mapped into the different channel indexes in $M$. For simplicity, we divide the $[2^t, 2^{t+1}], t \in [5, 10]$ into 10 scale bins and thus the total scale number $S = 60$. With the computed $b$, the value at coordinate $(\lfloor \frac{x}{N_s} \rfloor, \lfloor \frac{y}{N_s} \rfloor, b)$ can be defined as:

$$M(\lfloor \frac{x}{N_s} \rfloor, \lfloor \frac{y}{N_s} \rfloor, b) = 1,$$

where $N_s$ means the stride of the network. Encouraged by (Wu et al. 2018; Law and Deng 2018), to alleviate the difficulty of feature learning in the discrete distribution, we introduce the 2D Gaussian function to refine it. Given the radius $r = \lfloor \frac{b}{10} \rfloor$ and the host point $(x_h, y_h) = (\lfloor \frac{x}{N_s} \rfloor, \lfloor \frac{y}{N_s} \rfloor)$, the values of its neighbouring points can be formulated as:

$$M(x_i, y_i, b) = e^{-\frac{(x_i-x_h)^2+(y_i-y_h)^2}{2\sigma^2}},$$

where $(x_i, y_i)$ belongs to the neighbour set $\mathcal{N}(x_h, y_h, b)$ and $\sigma$ is set to 0.1 in our experiments. During training, the input
image is resized with the higher dimension equal to 256 and the loss of the scalemap training is a binary multi-class cross entropy loss:

\[ L_{scale} = -\frac{1}{|M|} \sum_i p_i \log \hat{p}_i + (1 - p_i) \log (1 - \hat{p}_i), \]

where \( p_i, \hat{p}_i \) are the ground truth label and prediction of the \( i \)-th pixel in \( M \). \(|M|\) indicates the pixel number in feature map \( M \).

For inference, given a threshold, we select all of the valid coordinates \((x_v, y_v, b_v)\) from \( M \) and compute the scale \( s_v = \max(h, w) \) by Eq. 1 based on its channel index \( b_v \). Finally, the scale proposal \([x_v, y_v, s_v, s_v]\) can be obtained.

**Scale adaptive soft-argmax operator**

As shown in (Wu et al. 2018), heatmap regression models can achieve better performance than coordinate regression models do. With this conclusion in mind, instead of regressing the keypoint coordinates, we detect them from the landmark response map. At the end of the backbone, the landmark response generation is used to generate a response map and compute the keypoint coordinates, we detect them from the landmark response map. At the end of the backbone, the landmark response generation is used to generate a response map and compute the keypoint coordinates. It only consists of one convolutional layer with kernel size 3. Instead of the argmax function, which is not differentiable, breaking the learning chain on neural networks (Luvizon, Tabia, and Picard 2017), we propose the scale adaptive soft-argmax operator which keeps the properties of specialized part detectors while being fully differentiable. Different from the usage in (Luvizon, Tabia, and Picard 2017) where it applies to the global response map cooperated with top-down methods, the proposed scale adaptive soft-argmax is performed on the scale aware locations cooperated with the bottom-up pipeline.

Given a scale proposal \( S = [x_1, y_1, x_2, y_2] \) where \( x_1, y_1, x_2, y_2 \) mean the top left and bottom right corner, we define the Softmax operation on it \( h \in \mathbb{R}^{H \times W \times K} \) as:

\[ \Phi(h_{i,j,c}) = \begin{cases} e^{h_{i,j,c}} / \sum_{m=x_1}^{x_2} \sum_{n=y_1}^{y_2} e^{h_{m,n,c}}, & (i,j) \in S, \\ 0, & 	ext{others} \end{cases} \]

where \( h_{i,j,c} \) is the value of heat map \( h \) at location \((i,j)\) of channel \( c \). The coordinates of the landmarks \( P_c = (\Psi_{c,x}, \Psi_{c,y}) \) corresponding to the \( S \) are given by:

\[ \Psi_{c,x} = \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{\mathbb{P}(i-x_1, w)}{w} \Phi(h_{i,j,c}), \]

\[ \Psi_{c,y} = \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{\mathbb{P}(i-y_1, h)}{h} \Phi(h_{i,j,c}), \]

where \( w, h \) indicate the width and height of \( S \). \( \mathbb{P}(x, y) \) returns \( x \) when \( 0 \leq x \leq y \) and 0 otherwise.

For keypoints regression based on each \( h \), we adopt the \( L_{keypoint} \) loss function as follows:

\[ L_{keypoint} = \frac{1}{2K} \sum_{c=1}^{K} \| \Psi_{c,*} - G_{c,*} \|^2, \]

where \( G_{c,*} \) means the ground truth keypoints. After obtaining the facial keypoints, the face boxes can be inferred by it conforming to the definition of us. Define the probability of a scale proposal as \( P_s \), the score of the corresponding face box is formulated as:

\[ P = P_s + \sum_{c=1}^{3} \max(\Phi(h_{s,*})), \]

where \( \max(\cdot) \) means the maximum operation on the spatial resolution and \( c \) from 1 to 3 represents the channels corresponding to left eye, right eye, and nose. We empirically utilize the keypoints information to weaken some false positive scale proposals.

**Backbone architecture**

We use two different CNN architectures as the backbones of KPNet, respectively. One is the robust stacked hourglass network and the other is DRNet with faster inference speed. The hourglass network only consists of one stacked hourglass and the channel number is reduced to 64. The first convolution layer with kernel size \( 7 \times 7 \), stride 2 is replaced by two small convolution layers without BN and ReLU between them. Kernel size \( 3 \times 3 \) with stride 2 and kernel size \( 3 \times 3 \) with stride 1 are assigned to them, respectively. Furthermore, we upsample the spatial resolution by a factor of 2 (using nearest neighbor upsampling for simplicity) at the end of the hourglass to reduce the bias caused by heavy downsample operations. Finally, the total stride of the hourglass is 2 and the parameter is \( \sim 1.04M \). Even though the network has become very lightweight after these specific modifications, abundant operations on large resolution feature maps still limit the inference speed of the network.

Following the principle of reducing the redundant operation on high-resolution feature maps, we design a simple and fast De-redundancy Net (DRNet). It can achieve \( \sim 5 \times \) faster
inference speed than the former hourglass with the same number of parameters. The details of DRNet are shown in Fig. 3. DRNet is a simple and lightweight network with only 11 layers. Given an input image, we first reduce it 4× via a 3×3 convolution layer with stride 2 and a 2×2 max pooling layer with stride 2. Then some residual blocks are followed and two nearest neighbor upsampling layers with factor 2 are used to upsample the spatial resolution. The total stride of DRNet is 2 and the large-span skip layer enables the network to retain as much input information as possible which makes it comfortable for low-resolution input. The lightweight structure without redundant operations on high-resolution feature maps ensures that it can achieve the faster inference speed.

Advantage insights of KPNet

All of the joint face detection and alignment algorithms (Zhang et al. 2016; King 2009) first generate the face boxes and then predict the corresponding landmarks. Limited by face detection accuracy which heavily relies on the tricky design of anchor boxes and is easily influenced by the ambiguous definition of bounding boxes, it’s hard to achieve better performance with faster speed. Compared with these top-down pipelines, KPNet has the following advantages. First of all, KPNet adopts the bottom-up mechanism that first locates the landmarks with the unambiguous definition, and then the face boxes can be inferred from keypoints. The precise definition of landmarks compared with face boxes allows it to easily achieve higher performance. Secondly, KPNet skips anchor designing and thusly is flexible to deploy this network without complicated design skills. Finally, different from the most face detection methods (Li et al. 2018; Liu et al. 2017), KPNet does not depend on high-resolution input. Through the combination of low-resolution input and lightweight network, it can achieve SOTA performance on both of the generic face detection (>20px) and alignment with fast inference speed (offline application with ∼1000fps).

Experiments

Implement details

We implement KPNet in PyTorch. Both of the hourglass and DRNet are randomly initialized under the default setting of PyTorch without pretraining on any external dataset. During training, we set the input resolution of the network to 256×256, which leads to an output resolution of 128×128. For training on generic face detection, we adopt the training set the same as (Liu et al. 2017) and none of the data augmentations is performed. For joint face detection and alignment, K is set to 5 representing the left eye, right eye, nose, left corner of the mouth and right corner of the mouth. We joint optimize the loss function \( L_{\text{scale}} \) and the \( L_{\text{keypoint}} \) with weightloss 1:1 via SGD. Due to the huge pixels in scalemap \( M \in \mathbb{R}^{128\times128\times60} \), the weight of \( L_{\text{scale}} \) is set to 10000 for faster convergence. Benefiting from the low-resolution input and lightweight backbone, we use a batch size of 128 and train the network on 4 GTX 1080Ti GPUs. For training on AFLW, because of the missing annotation of some face boxes in the training set, we only train the \( L_{\text{keypoint}} \) for the annotated facial landmarks. K is set to 19 to correspond to the annotated facial landmarks in AFLW. We adopt the data augmentation strategy the same as (Feng et al. 2018) for preventing over-fitting. We train the network for 150k iterations with a learning rate warm-up strategy. The learning rate is linearly increased to 0.01 from 0.00001 in the first 50k iterations and we reduce it to 0.001 for the last 50k iterations.

At the inference stage, we first generate the scale proposals through the predefined threshold from scalemap, and then compute the corresponding keypoints via then scale adaptive soft-argmax according to Eq. 5 and Eq. 6. Finally, NMS with IOU 0.6 is adopted on the face boxes inferred from these keypoints.

Test benchmarks

We evaluate KPNet on the generic face detection benchmarks FDDB (Jain and Learned-Miller 2010), AFW, MALF, and face alignment benchmark AFLW (Koestinger et al. 2011). We adopt the relabeled version provided by (Liu et al. 2017) where some missing faces are re-annotated. We follow (Feng et al. 2018) to adopt the AFLW-Full in our experiments where 20,000 and 4,386 images are used for training and testing, respectively. For face detection, we follow the protocol of (Liu et al. 2017).

<table>
<thead>
<tr>
<th>Network</th>
<th>FDDB (%)</th>
<th>AFW (%)</th>
<th>MALF (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA(_{\text{base}})</td>
<td>96.0</td>
<td>100.0</td>
<td>96.49</td>
</tr>
<tr>
<td>DRNet</td>
<td>96.6</td>
<td>99.6</td>
<td>97.1</td>
</tr>
<tr>
<td>Hourglass(_{\text{tight}})</td>
<td>96.7</td>
<td>99.8</td>
<td>97.59</td>
</tr>
</tbody>
</table>

Table 1: Recall on FDDB, AFW, and MALF. All of the results are evaluated on the top 100 proposals.

Ablation study

Fine-grained scale approximation. Fine-grained scale approximation is a key component of KPNet. To understand its performance, we directly evaluate its recall on FDDB, AFW, and MALF. Similar to the evaluation metric in Sec., we compute the recall of the top 100 scale proposals. Furthermore, we implement the anchor-based detector RSA\(_{\text{base}}\) (Liu et al. 2017), which is the SOTA algorithm on generic face detection. According to their claimed configuration, we evaluate the recall for comparison with the same protocol. The result is shown in Tab. 1. Hourglass\(_{\text{tight}}\) is the modified hourglass network in Sec. The fine-grained scale approximation can achieve a comparable recall to the SOTA anchor-based algorithm without relying on the experience design.

Advantage of scale adaptive soft-argmax operator. We introduce the scale adaptive soft-argmax (SS) to predict the facial keypoints coordinates from the heatmap. In order to better evaluate the superiority of SS over coordinate regression and argmax, We conduct different experiments with DRNet. For coordinate regression, we replace the SS by a fully connected layer to directly regress the keypoints coordinates. We adopt the global average pooling on
Table 2: Performance on different test sets. We report the recall at false positive number 50. All of the experiments except RSA_{base} are based on DRNet. The two values in PFDDDB mean the evaluation on IOU 0.7 and 0.8, respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FDDDB</th>
<th>FDDDB_{90°}</th>
<th>FDDBB_{90°}</th>
<th>PFDDDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA_{base} regression</td>
<td>92.15%</td>
<td>57.67%</td>
<td>56.78%</td>
<td>92.84/59.93</td>
</tr>
<tr>
<td>argmax</td>
<td>90.86%</td>
<td>68.96%</td>
<td>67.2%</td>
<td>90.94/47.97</td>
</tr>
<tr>
<td>SS</td>
<td>88.98%</td>
<td>50.65%</td>
<td>49.9%</td>
<td>83.51/43.9</td>
</tr>
<tr>
<td></td>
<td>91.6%</td>
<td>69.32%</td>
<td>69.97%</td>
<td>91.29/61.44</td>
</tr>
</tbody>
</table>

Table 3: Ablation study for error analysis. The recall values are evaluated at false positive number 50 and 1 on FDDDB. SP, SS and GT box mean the scale proposal, scale adaptive soft-argmax and ground-truth face boxes.

<table>
<thead>
<tr>
<th>Backbone</th>
<th>SP</th>
<th>SS</th>
<th>GT box</th>
<th>FDDDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRNet</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>90.7/47.3</td>
</tr>
<tr>
<td>DRNet</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>91.6/81.1</td>
</tr>
<tr>
<td>DRNet</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>96.6/96.6</td>
</tr>
<tr>
<td>Hourglass_{light}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>91.64/12.52</td>
</tr>
<tr>
<td>Hourglass_{light}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>92.61/80.9</td>
</tr>
<tr>
<td>Hourglass_{light}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>96.72/96.72</td>
</tr>
</tbody>
</table>

Table 4: A comparison of different approaches in terms of the average error normalized (×10^{-2}) on AFLW.

<table>
<thead>
<tr>
<th>Methods</th>
<th>average normalized error</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCD-CNN (Kumar and Chellappa 2018)</td>
<td>2.40</td>
</tr>
<tr>
<td>TSR (Lv et al. 2017)</td>
<td>2.17</td>
</tr>
<tr>
<td>LAB (Wu et al. 2018)</td>
<td>1.25</td>
</tr>
<tr>
<td>SAN (Dong et al. 2018a)</td>
<td>1.91</td>
</tr>
<tr>
<td>PFDD 1X (Guo et al. 2019)</td>
<td>1.88</td>
</tr>
<tr>
<td>Wing (Feng et al. 2018)</td>
<td>1.65</td>
</tr>
<tr>
<td>CPM+SBR (Dong et al. 2018b)</td>
<td>2.14</td>
</tr>
<tr>
<td>KPNet+DRNet</td>
<td>1.87</td>
</tr>
<tr>
<td>KPNet+Hourglass_{light}</td>
<td>1.45</td>
</tr>
</tbody>
</table>

The results are shown in Tab. 2. All of the models are trained on normal faces without rotation augmentation. SS performs better than others, even the anchor-based RSA_{base} with high-resolution input and image pyramid, KPNet can achieve comparable performance, even more, robust in face-rotation scenarios. In actually, detecting face boxes from key points is important for the lightweight network than directly regressing bounding boxes. Key points have less uncertain information which is easier for lightweight models to fit. Besides, the semantic information of the landmarks is fixed, even if the face angle/pose changes. This ensures its robustness to face angle/pose variance.

**Error analysis.** KPNet simultaneously outputs fine-grained scale heatmap and landmarks response map. To understand how each part contributes to the final error, we perform an error analysis by replacing the predicted scale proposal with the ground-truth boxes. Furthermore, to evaluate the SS contribution to KPNet, we detect the faces only by scale proposal.

**Comparisons with SOTA algorithms**

For face detection, we compare our KPNet with state-of-the-art methods (Liu et al. 2019; Li et al. 2018; Zhang et al. 2017a; Tang et al. 2018; Zhang et al. 2017b; Wang et al. 2017b; Zhang et al. 2017c; Li et al. 2015; Liu et al. 2017; Yang et al. 2015; Yu et al. 2016; Mathias et al. 2014; Chen et al. 2016; Farfade, Saberian, and Li 2015; Yang et al. 2014; Zhang et al. 2016) and the DLIB c++ library (King 2009), which supports for joint face detection and alignment. Fig. 4 shows the comparison with other approaches on three benchmarks. On AFLW, our algorithm KPNet can achieve 99.53% AP and 98.72% AP by Hourglass_{light} and DRNet, respectively. On FDDDB, KPNet+Hourglass_{light} recalls 92.61% faces with 50 false positives as shown in Fig. 4a which outperforms most of the approaches. On MALF, our methods can also achieve a comparable result with the state-of-the-art. It should be noticed that the shape and scale definition of the bounding box on each benchmark varies. KPNet can be easily applied to these benchmarks without complicated design choices.

For face alignment, we compared KPNet with other state-of-the-art methods on AFLW. AFLW is a challenging dataset that has been widely used for evaluating face alignment algorithms. As shown in Tab. 4, our KPNet+Hourglass_{light} outperforms all of the other approaches and KPNet+DRNet can also achieve comparable performance.

Furthermore, we compare KPNet+DRNet with the bottom-up associative embedding (AE) (Newell, Huang, and Deng 2017; Law and Deng 2018) and top-down RPN+DRNet. As shown in Tab. 6, whether adopting anchor-based RPN or AE, the performance is strictly limited by the capacity of lightweight backbone. It’s in stark contrast that KPNet with fine-grained scale approximation and scale adaptive soft-argmax achieve excellent performance. To better understand the performance of KPNet, we visualize some images sampled from FDDB and AFLW in Fig. 5.
Figure 4: Comparison to the state-of-the-art on face detection benchmarks. The proposed KPNet with low-resolution input and lightweight architecture can achieve comparable results with other well-designed anchor-based algorithms.

Figure 5: Visualization of joint face detection and alignment. Images are sampled from FDDB and AFLW.

Table 5: Comparison with different approaches in terms of the inference speed and performance on FDDB and AFLW. The online speed means we evaluate it with batch size 1 and the offline speed means we evaluate it with batch size (> 32). W/o means this application is not supported (e.g., $S^2AP$ can only support the batch size 1 due to its high-resolution input.). $T_i$, $P_{100}$, $X_P$ and $X$ indicate the GTX 1080Ti, NVIDIA P100, TITAN X Pascal and TITAN X GPU. The * in MTCNN means this result is normalized by inter-ocular distance and other results on AFLW are normalized by face size.

Table 6: Comparison with top-down method and bottom-up methods AE proposed in pose estimation.

Analysis of the inference speed

In this section, we explore the performance and speed in detail compared with other approaches as shown in Tab. 5. We report the recall at false positive number 50 on FDDB and the NME ($\times 10^{-2}$) on AFLW.

In the offline applications, KPNet with DRNet can achieve $\sim 1000$ fps at GTX 1080Ti, faster than other face detectors with a large margin. Even compared with the state-of-the-art algorithms on face alignment, KPNet still has a faster model inference speed.

Both the MTCNN and DLIB are the popular frameworks for joint face detection and alignment. KPNet outperforms them with a large margin in terms of inference speed and performance. No complex hyperparameter designing is required so that it can be easily applied to different scenarios.

Conclusion

This paper proposes a simple, lightweight but accurate framework KPNet which does away with anchor boxes. It focuses on joint generic face (> 20px) detection and alignment. Unlike most face detection methods and top-down joint face detection and alignment methods, KPNet adopts the bottom-up mechanism. It first predicts the facial landmarks from a low-resolution image via the well-designed fine-grained scale approximation and scale adaptive softmax operator. Finally, the precise face bounding boxes, no matter how we define it, can be inferred from the landmarks. KPNet can effectively alleviate the vague definition of the face bounding box. Without bells and whistles, KPNet achieves state-of-the-art accuracy on generic face detection and alignment benchmarks with only $\sim 1$M parameters. The model inference speed can achieve $\sim 1000$ fps on GPU and it’s easily deployed to most modern front-end chips.