BEE POSE ESTIMATION FROM SINGLE IMAGES WITH CONVOLUTIONAL NEURAL NETWORK

Le Duan[1]*, Minmin Shen*[1]*, Wenjing Gao*[2]*, Song Cui*[2*]*, Oliver Deussen*[1]

INCIDE center, University of Konstanz[1], Germany
Institute of High Performance Computing[2], Singapore

ABSTRACT

In this paper, we present a deep convolutional neural network (ConvNet) based framework for estimating the bee pose from a single image. Unlike some existing human pose estimation methods that localize a fixed number of body joints, our method handles the cases with a varying number of targets. Compared to the existing bee pose estimation methods, our framework is more robust and accurate. It is effective even for some challenging images (e.g., when the bee is fed sugar water with a stick). The proposed framework learns a mapping from the global structure and local appearance of a bee to its pose. We evaluated our method on two challenging datasets. Experiments showed that it has achieved significant improvements over the existing insect pose estimation algorithms.

Index Terms— Insect pose estimation, ConvNet

1. INTRODUCTION

Animal pose estimation is important for behavior study of animals such as bees. In Fig. 1, our dataset shows that behaviors of bees could be trained in controlled stimulus conditions, e.g., different light conditions or human interference such as feeding the bee sugar water with a stick (Fig. 1b). These behaviours can be reflected as movements of bee body parts such as their antennae or mouthparts. Bee pose estimation is challenging because the bee body parts exhibit self-similarity and self-occlusions [1]. Moreover, the number of bee body parts could be varying (Fig. 1c-f) and there may be weak correlation among the movements of different bee body parts.

In the videos to be analyzed, a stick is used to feed the bee with sugar water and the bee is responding by extending its tongue. Thus, we need to localize the tips of bee's antennae and tongue, and to distinguish the sugar stick from the bee body parts. In [2], a random forest (RF) based framework is proposed to address this issue. The approach in [2] uses RF model as the classifier and the global structure of bees is represented by geometric features (e.g., edge histogram). When the frames contain no stick, the performance of the algorithm has comparably precision to human annotators. However, the RF classifier is not able to distinguish the stick from bee body parts and the geometric features may not represent the bee global structure well (e.g., Fig. 1c and Fig. 1f may have similar geometric features).

Nowadays, deep convolution neural networks (ConvNets) are widely used in computer vision. Existing networks such as GoogLeNet [3] and VGGNet [4] are proven to have better performance compared to other methods in pose estimation [5] and image classification [6]. In this paper, we use a ConvNet based algorithm for bee pose estimation. Methods that are proposed for human or hand pose estimation are not suitable in our task. These methods assume strong relationships between the joints [5] and the number of joints is fixed. In our study, the bee body parts have weak correlations and parts are often missing under certain experimental conditions. In addition, data used for hand pose estimation is often depth image, which contains additional depth information [7] while our dataset only contains 2D images.

To address the aforementioned issues, we present a unified framework that utilizes ConvNets for bee pose estimation. The contribution of our work is two-fold:

1. As far as we know, our framework is the first method that uses ConvNet for bee pose estimation. We present a new ConvNet architecture based on VGGNet for multiple landmark localization.

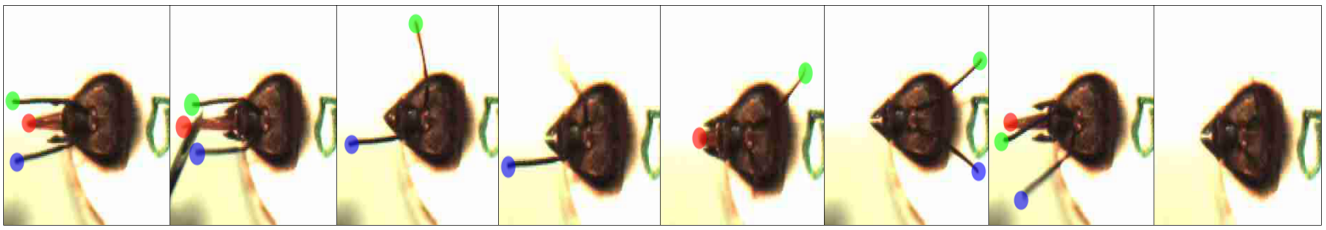
2. Experimental results show that our method outperforms the existing bee pose estimation framework. For example, the proposed method has reduced the tip position error by more than 42% for a dataset with shadows and background noises present.

2. METHODOLOGY

In this paper, our objective is to estimate the bee pose $\mathbf{P} = \{\mathbf{x}^n | 0 \leq n \leq N, \forall \mathbf{x}^n \in \Re^2\}$ from an image I, where $\mathbf{x}^n = \{x^n, y^n\}$ denotes the position of a tip in image coordinate system. We formulate this problem as finding the maximum posteriori of a pose given an image I, i.e., $p(\mathbf{P}|I)$, which can be approximated as:

$$p(\mathbf{P}|I) \propto p(I|\mathbf{P})p(\mathbf{P}), \tag{1}$$

*CORRESPONDING EMAIL: SONGCUI@ACM.ORG



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 1. Example images of various bee poses. The right antenna is represented by green dot, the tongue is represented by red and the left antenna is colored in blue. (a) all tips are present; (b) The sugar water is fed to a bee with a stick; (c)-(e) some body parts are not visible; (f) the antennae may move backwards in some rare cases; (g) part of the tongue is occluded by the right antenna; (h) all parts are absent.

where $p(I|\mathbf{P})$ is the probability of the image I for a particular pose $\mathbf{P}$, and the $p(\mathbf{P})$ refers to the pose confidence map. We present a ConvNet based framework for solving Eq. (1).

ConvNets consists of several types of layer architectures such as convolutional, pooling, ReLU, and fully connected layers. They are cascaded in sequence. They are efficient in learning hierarchical feature representations of input data. General image features are captured by lower layers and specific features and information only relevant to the input dataset are captured by higher layers. In our framework, we use ConvNets to learn $p(\mathbf{P})$ and $p(I|\mathbf{P})$ from training data. On one hand, since the architect of VGGNet is relatively interpretable, we modify its structure to generate confidence map of possible positions of each tip (Fig. 2b-c), i.e., $p(\mathbf{P})$. On the other hand, we employ a fine-tuned GoogLeNet for image feature extraction and constructing a feature space. GoogleNet is used for learning the representation of global structure instead of VGGNet since it has more complex architect and more powerful representation capability. We find the K nearest neighbours (KNN) of the data point representing the testing image and compute the probability masses based on the KNN result (Fig. 2d-f), which can be regarded as $p(I|\mathbf{P})$. Combining $p(\mathbf{P})$ and $p(I|\mathbf{P})$, the tip positions can be estimated. The overall framework is shown in Fig. 2.

2.1. Confidence map generation

For confidence map generation, the VGGNet is used as the base model. All the tips share the weights of lower layers. Feature maps of individual tips and background are learned from the corresponding layers (e.g., Fig. 3 fonv_tip1 layer for tip 1 and fconv_bg for background, etc). Layers for extracting feature maps of each tip as well as the background are independent since the movements of each tip do not correlate with the others. Fig. 3 gives a detailed view of our modified VGGNet. The feature map of each tip gives detailed information of insect pose and tip appearance, which can be regarded as the response of filters for a specific tip of the whole image. The final confidence map of each tip is predicted from the corresponding feature map through 1×1 convolution (e.g., Fig. 3 fconv_tip1_1 layer) across feature map. The confidence map has four channels. The first three channels correspond to the three tips and the last channel corresponds to the background. The size of each channel is 44×44 and resizing the confidence map to the size of input image is required. Fig. 2(c) shows an example of imposed confidence map, green/blue and red regions indicate the possible locations of right/left antenna and tongue, respectively.

2.2. Confidence refinement

The GoogLeNet was originally trained on ImageNet [8] and it can not fully represent image characteristics of our dataset. To transfer the representation of ImageNet to our dataset, we seek to localize the two antennae as an auxiliary task. After fine-tuning, the Euclidean distance between the prediction and labels is minimized, and the fine-tuned GoogLeNet is used for extracting features representing the whole image. To fine-tune the model, the lowest three convolutional layers ("conv1/7×7_s2", "conv2/3×3_reduce", "conv2/3×3") are frozen without tuning, and the remaining layers are re-trained on our dataset. The weights of the net are initialized by the pre-trained model trained on ImageNet. Features of an image are extracted from the 'pool5' layer, and the dimension of the feature vector is 1024.

Assuming that similar poses would have similar features, the data point of the test image should lie close to the training images with similar pose in the feature space. As illustrated in Fig. 2(a-e), features of a test image is extracted by the fine-tuned GoogLeNet. Then the K nearest neighbors of that feature are found in the feature space. The corresponding training subset $\mathbf{S}$ of KNN result can be regarded as having similar pose as the test image. We further assume that the tips locations take the form of Gaussian distribution, the tips' probability masses of $\mathbf{S}$ is computed to form new distributions of tips locations(Fig. 2f).

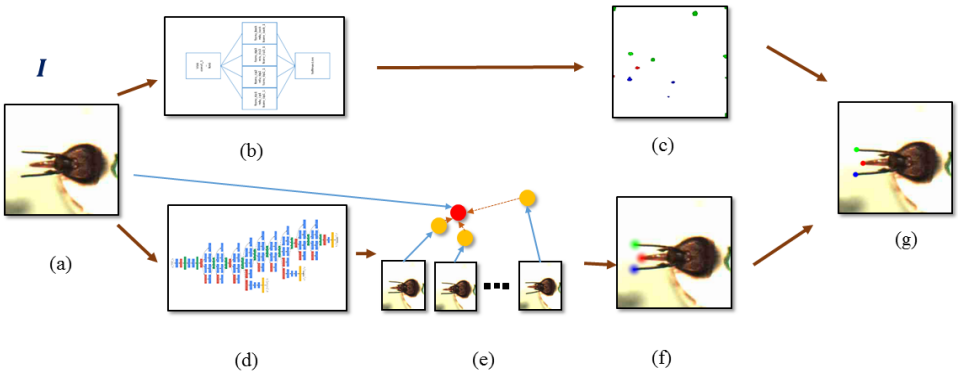


Fig. 2. The flowchart of the proposed framework. (a) Input image; (b) Modified VGGNet; (c) Confidence map; (d) Fine-tuned googLeNet; (e) KNN search; (f) Probability mass of KNN result; (g) Final tip localization.

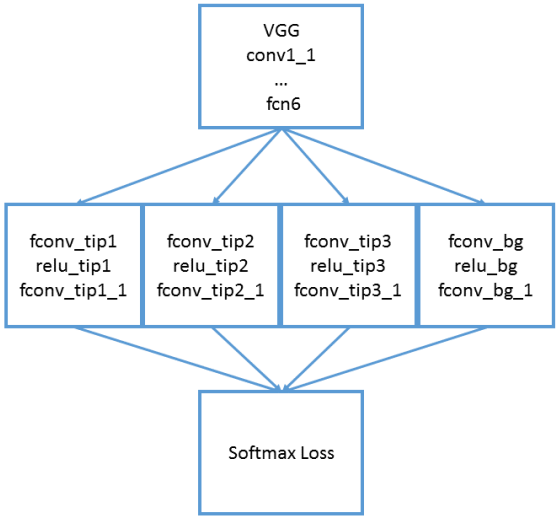


Fig. 3. Details of the modified VGGNet in our framework. All the tips share weights up to fcn6 layer, features for specific tips and background are learned from corresponding path.

2.3. Localization

Confidence maps provide probabilities of each tip position. Due to the visual similarity of tips and background, the confidence maps may contain noise. In addition, we cannot detect the absence of tip(s) from the confidence map only. The distribution of **S** can provide additional information of possible locations. Moreover, the absence of tip(s) can be detected by the number of absent tips in **S**, i.e., if the number of absent tips of right antenna from **S** is greater a threshold t, the right antenna is absent in the input image.

In Eq. (1), the posterior probability $p(\mathbf{P}|I)$ is approximated as the product of the likelihood of the image evidence given a particular pose and pose confidence probability map of that image. $p(I|\mathbf{P})$ is the distribution of **S** and $p(\mathbf{P})$ corresponds to the confidence map. Finally, mean shift algorithm [9] with a flat kernel is used to locate the tip positions.

3. TRAINING

3.1. Data preparation

We randomly selected 8705 images as training data, which contains frames from several types of bee videos. It is noted that the number of tips in those images is not fixed, i.e., some images may not have tongue and others may have missing tips due to blur images or variation of lighting conditions. To train our VGG-based net, we further split the images into training set and validation set. The training set contains 7224 images and the remaining 1481 images are used as validation images.

For the training set, we applied rotation (from -165 to 180 degree with step size of 15 degrees) and flipping, leading to 347788 images in the final dataset. Since the number of images with bee antenna going backward (Fig. 1f) is less than others, we augmented training data to balance the number of types of images, and the final training set is composed of 9109 images for fine-tuning the GoogLeNet.

3.2. Training details

To train the VGG-based network, the estimation of the bee tip locations is considered as a classification problem. Each pixel

Table 1. Performance comparison on Dataset A.

Algorithms	Left antenna			Tongue			Right antenna		
	pos. error	FN(%)	FP(%)	pos. error	FN(%)	FP(%)	pos. error	FN(%)	FP(%)
Proposed	5.6	3	0	7.9	22	2	4.3	1	0
RF-based method	13.8	5	0	13.6	8	23	8.5	5	0

Table 2. Performance comparison on Dataset B.

Algorithms	Left antenna			Right antenna		
	pos. error	FN(%)	FP(%)	pos. error	FN(%)	FP(%)
Proposed	10.2	4	1	7.1	4	2
RF-based method	14.3	2	1	7.0	4	2

is assigned a class label tensor and the training label for the input image is of size $42 \times 42 \times 4$. The first 3 channels represent 3 tips (c1:right antenna, c2: left antenna, c3: tongue) and the last channel is the representation of background. The values of the first three channels are all zero if the corresponding tip(s) is/are missing.

4. EXPERIMENT

4.1. Evaluations

We evaluated the performance of our proposed method on two datasets described in Fig. 1. The two datasets are collected by our biological partner during a behavioral experiment under different experimental settings such as various light conditions and video frame rates. Dataset A contains 2788 testing images and Dataset B contains 9003 testing images.

Results from our method are compared with the results from the RF-based method [2] and the ground truth, which are generated by human annotators. We compute false positive (FP) and false negative (FN) rates to evaluate the effectiveness of our method, e.g., a FN of the left antenna indicates that the left antenna is predicted to be absent by our algorithm but it is actually present. The average Euclidean distance in pixels between estimated tip positions and their true positions is used to evaluate localization precision.

4.2. Implementation details

For the KNN search, K is set to 30, and the threshold t for detecting the absence of the tip is set to 20. The mean shift kernel bandwidth is fixed at 0.05 for all images.

The lower layers of our VGG-based net used pre-trained weights as initialization and are fine-tuned with an initial learning rate of 0.001. The other layers are randomly initialized with an initial learning rate of 0.01. All the layers are fine-tuned together. The base learning rate for fine-tuning the GoogLeNet is set to 0.005, and the learning step size is set to 32000. The maximum number of iteration is set to 5000000.

4.3. Results and discussion

We compare the performance of our method and the RF-based method on two datasets. As the images in Dataset B contain no tongue, we only compare the results for two antennae.

Table 1 shows the comparison results of two methods on Dataset A, which is considered as a more challenging task due to the complex background: some shadows and background noise are present. It is even confusing for humans to determine the presence of the tongue in some cases. For example, only a small part of the tongue is visible in some images (Fig. 1e), and the antenna and tongue are overlapping in the other cases. We can see that our method produces less pixel errors than the RF-based method for all the tips and less FN rate for the antennae. Our method produces a high FN rate (22%) for the tongue because when the tongue is too short (Fig. 1e), the KNN search results of our method indicate that the input image is similar to those without the tongue. On the contrary, the RF-based method is able to detect the presence of short tongue, but it also provides high FP rate (23%).

In Dataset B, the images without the sugar stick have a clear background while some shadows are present with the sugar stick in some images. Table 2 compares the results of two methods on Dataset B. Our method performs better than the RF-based method in nearly every metric except the FN rate for left antenna. This is due to the high FN when the antenna is not obvious (Fig. 1d), and the KNN results found the images without the left antenna. The average pixel errors of both methods are rather small compared to the size of a bee head (120×160 pixels).

5. CONCLUSION

In summary, we present a new algorithm based on ConvNet for estimating a bee pose from a single image. The proposed framework utilizes the powerful representation capability of ConvNet to learn the mapping from the local appearance and global structure of a bee to its corresponding pose. Our method is able to localize a varying number of targets in complex background, especially for the cases when the bee is fed sugar water with a stick. It has been shown that our method outperforms the existing bee pose estimation algorithm on two challenging datasets of bees.

6. REFERENCES

[1] Minmin Shen, Paul Szyszka, C Giovanni Galizia, and Dorit Merhof, "Automatic framework for tracking honeybee's antennae and mouthparts from low framerate video," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 4112–4116.

[2] Minmin Shen, Le Duan, and Oliver Deussen, "Single-image insect pose estimation by graph based geometric models and random forests," in *ECCV workshop on Bioimage Computing*, 2016, pp. 217–230.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[4] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang, "Structured feature learning for pose estimation," *arXiv preprint arXiv:1603.09065*, 2016.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit, "Hands deep in deep learning for hand pose estimation," *arXiv preprint arXiv:1502.06807*, 2015.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[9] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.