Abstract

Deep neural networks achieve good performance in many computer vision problems such as face alignment. However, when the testing image is challenging due to low resolution, occlusion or adversarial attacks, the accuracy of a deep neural network suffers greatly. Therefore, it is important to quantify the uncertainty in its predictions. A probabilistic neural network with Gaussian distribution over the target is typically used to quantify uncertainty for regression problems. However, in real-world problems especially computer vision tasks, the Gaussian assumption is too strong. To model more general distributions, such as multimodal or asymmetric distributions, we propose to develop a kernel density deep neural network. Specifically, for face alignment, we adapt state-of-the-art hourglass neural network into a probabilistic neural network framework with landmark probability map as its output. The model is trained by maximizing the conditional log likelihood. To exploit the output probability map, we extend the model to multi-stage so that the logits map from the previous stage can feed into the next stage to progressively improve the landmark detection accuracy. Extensive experiments on benchmark datasets against state-of-the-art unconstrained deep learning method demonstrate that the proposed kernel density network achieves comparable or superior performance in terms of prediction accuracy. It further provides aleatoric uncertainty estimation in predictions.

1. Introduction

Face alignment, or facial landmark localization, is a fundamental step for facial behavior analysis such as face recognition, facial expression estimation, and head pose estimation. Classical work on face alignment mainly adopts a cascade regression framework based on local image features [50, 49, 48, 47], which is sensitive to initialization and achieves limited performance on challenging dataset. With the introduction of deep learning based methods for regression and feature representation learning, the state-of-the-art accuracy in face alignment is achieved [41, 52, 43, 3, 11, 46]. However, the performance of face alignment remains sensitive to face image quality. Challenges such as large head pose, object occlusion or low resolution may lead to poor landmark detection results. Moreover, existing deep learning based methods are susceptible to small image perturbations such as an adversarial attack, which may result in a large difference in the prediction. More importantly, existing deep learning methods cannot predict their output uncertainty. Therefore, it is important to develop a probabilistic deep neural network to quantify the prediction uncertainty and to avoid making over-confident wrong decisions.

To these goals, we propose a Kernel Density Deep Neural Network (KDN). Different from the deterministic approach that gives a point estimation for each input, our model outputs target probability distribution for each input. Moreover, rather than assuming the output follows a Gaussian distribution, the proposed method can capture more general probability distribution, such as multimodal or asymmetric distribution. With the target probability distribution, we can quantify the prediction confidence to distinguish challenging input image caused by large head pose, object occlusion or low resolution and to identify landmarks under occlusion. To further exploit the output probability map, we further extend our model to multi-stage cascade framework so that the probability map produced in the last stage can serve as input to guide the detection in the next stage to progressively improve the landmark detection accuracy.

The contributions of work are summarized as follows: 1) We introduce the Kernel Density Deep Neural Network that produces target probability map, without assuming a specific parametric distribution. The probability map can be used to quantify the uncertainty of the output and to identify the challenging landmarks. And we further extend our model to multiple stages to use the output probability map to progressively improve the landmark detection. 2) We show that the estimated uncertainty in our method can be used to detect occluded landmarks without occlusion supervision. 3) We show the proposed method can be generally extended to other regression tasks such as action unit intensity estimation.
2. Related work

2.1. Probabilistic Neural Network

To quantify uncertainty in neural networks, the probabilistic neural network was proposed to model the conditional target probability distribution given its input and neural networks are used to predict the parameters of the probability distribution. For regression tasks, it is often assumed that the target follows Gaussian distribution [29, 21, 19]. And neural network is used to predict the mean and variance for the Gaussian distribution. In this way, the prediction is given by the mean and the uncertainty is quantified by the variance.

However, for many real-world problems, the target distribution may be more complicated, e.g. asymmetric or multimodal, which the Gaussian distribution cannot adequately capture. To deal with this issue, one way is to parameterize a different distribution suitable for the specific problem. For instance, [28] used a Gamma distribution to model the distribution of surgery duration. And [34] used a von Mise distribution to model the distribution of object pose. Another way is to use a mixture distribution which is more flexible in approximating distributions with different shapes. For instance, [1] extended the von Mises distribution to a mixture of von Mises distribution to handle multimodal distribution. These methods typically still have assumptions and are not generally applicable.

We are interested in modeling the distribution of landmark location, which has high probability near the boundary of the facial parts. Therefore the distribution typically does not follow some standard parametric distribution such as Gaussian. And different landmarks may have very different distribution shape.

2.2. Face Alignment

Face alignment is typically treated as a regression problem where given a face image, it aims to localize certain facial key points in the image. Classic methods lie in the categories of Active Shape Model (ASM) [26], Active Appearance Model (AAM) [10, 18, 25, 37], Constrained Local Model (CLM) [20, 38] and Cascade Regression [7, 4, 54, 5, 49]. ASM models the statistical shape of objects, while AAM models both the shape and the appearance features. CLM is similar to AAM that models shape prior using principal component analysis (PCA) that projects both local appearance features and shape features onto the bases. Cascade Regression refines landmark localization stage by stage. These classic methods rely on hand-crafted local image features and are usually sensitive to initializations. They are outperformed by deep learning based methods which use deep feature representation.

Deep learning based method for face alignment was first proposed in [41] and achieved better performance than classical methods. Later on, more works on face alignment using deep learning framework has been explored [53, 52, 43, 12] but they are all based on coordinate regression.

Until recently, fully convolutional neural network (FCN) [23] based methods established new state-of-the-art for face alignment and body pose estimation [42, 27, 45]. And most of these face alignment methods [3, 46] follow the architecture of Stacked Hourglass [27]. The stacked modules refine the network predictions after each stack, similar to the idea of Cascade Regression. Instead of directly predicting the landmark coordinates, it predicts a heatmap with same size as the input image and the landmark location is predicted by the coordinate on the heatmap with largest response. The idea of the heatmap based regression is similar to a fully convolutional neural network which preserves the spatial information of the input image and reduce the parameters brought by fully connected layers.

2.3. Fully Convolutional Network Loss

State-of-the-art face alignment methods adopt FCN structure with a heatmap regression loss. The loss function is typically defined as the mean squared error between the predicted heatmap and the ground truth heatmap. This loss function is originally introduced and widely used in human pose estimation [42, 27, 2, 8, 6]. Besides this typical loss function, there are several other options in literature introduced for solving other problems such as image segmentation.

One choice is to treat the problem as a multi-class classification problem where each pixel location in the heatmap corresponds to one class and use the softmax cross entropy loss over the 2D heatmap. This was used in Mask-RCNN [16] for human body joint estimation. One pixel location with highest probability in the heatmap is selected as the estimation. Solving a regression problem using softmax cross entropy loss also exists in other problems such as face age estimation [31], where we have discrete age labels. These age labels, though discrete, are not independent because labels with close values should be more confusing with each other in classification. Therefore, to some extent, this loss function abandons part of the information provided by the label values. To address this issue, the paper [31] further uses L2 loss of the mean computed from the softmax probability. This idea has been explored in other tasks such as body pose estimation [40], headpose estimation [35]. And [13] further proposes to use L1 loss instead of L2 loss in face age estimation, while [44] proposes to apply wing loss to heatmap regression that is first proposed in [12] for traditional coordinate regression in face alignment. These works, while achieving satisfactory performance in terms of low prediction error, cannot accurately quantify prediction uncertainty. More recently, [15] proposes to estimate covariance matrix.
besides the mean of a multivariate Gaussian distribution, thus incorporating uncertainty into this framework. All the aforementioned works that add L1, L2 or Gaussian negative log likelihood losses besides the heatmap loss assumes the output distribution as single-modal, implicitly or explicitly. However, we show in this paper that this is often not the case in real-world computer vision problems. Therefore using the mean for inference will lead to erroneous predictions for multimodal cases.

Another choice is to treat the problem as a classification problem for each pixel [32, 17, 33]. There are two types of classification, one is binary classification, where each pixel will be classified as the target or not. And the ground truth binary label is created by assigning 1 to all the pixels in a certain neighborhood around the ground truth target pixel location and 0 to the rest pixels in the heatmap. The other choice is multi-class classification [9], where each pixel is classified as either one of the body or facial part regions or the background. These loss functions are used very often for segmentation tasks. And since it usually uses a softmax or sigmoid cross entropy loss which is also the negative log likelihood of a Categorical distribution, it is able to quantify classification uncertainty. However, they do not achieve as good performance as the previous loss functions as studied in [40] and it is difficult to define the ground truth body or facial part regions given only a single pixel location as the ground truth keypoint location.

Therefore, in this work, we propose a different loss function and a corresponding inference method that achieves state-of-the-art performance and provide good aleatoric uncertainty estimation. Our work differ from previous works in terms of explicitly quantifying uncertainty in fully-convolutional based architecture without adding additional fully connected layers to predicted covariance as [15].

3. The Proposed Method

Our method is built on the probabilistic neural network framework. We assume target y (landmark coordinates) is a random vector that follows \( p(y \mid x; \Theta) \), where \( x \) is the input image and \( \Theta \) is the neural network parameter. And \( p(y \mid x; \Theta) \) is parameterized by the neural network output.

3.1. Kernel Density Network

Instead of assuming the target follows Gaussian distribution as being done by current models, we propose to model the target probability with multi-variate kernel density function [39] in order to capture more general probability distributions, including multimodal and non-symmetric distributions.

Denote \( m, n \) as the height and width of the output \( \pi(x; \Theta) \) from the Hourglass module, \( \pi_{ij} = [i, j]^T \) as the pixel location in the output map, where \( 1 \leq i \leq m, 1 \leq j \leq n \). According to the multivariate kernel density distribution [39], the target distribution can be expressed as

\[
p(y \mid x; \Theta) = \sum_{i=1}^{m} \sum_{j=1}^{n} K_{\Sigma}(y - \mu_{ij}) \pi_{ij}(x; \Theta) \tag{1}
\]

where \( K_{\Sigma}(y - \mu_{ij}) \) is a Gaussian kernel whose value is the standard 2D Gaussian’s probability density at \( \Sigma^{-\frac{1}{2}}(y - \mu_{ij}) \) normalized by \( |\Sigma|^{-\frac{1}{2}} \), i.e. \( K_{\Sigma}(y - \mu_{ij}) = |\Sigma|^{-\frac{1}{2}} \phi(\Sigma^{-\frac{1}{2}}(y - \mu_{ij})) \). \( \pi(x; \Theta) \), the output of the neural network, is a weight map of the dimension \( m \times n \), where each pixel value \( \pi_{ij}(x; \Theta) \) represents the weight of the Gaussian kernel \( K_{\Sigma}(y - \mu_{ij}), 0 \leq \pi_{ij}(x; \Theta) \leq 1 \) and \( \sum_{i=1}^{m} \sum_{j=1}^{n} \pi_{ij}(x; \Theta) = 1 \).

Thus we form a continuous probability \( p(y \mid x; \Theta) \) based on the Gaussian kernels \( K_{\Sigma}(y - \mu_{ij}) \) and their corresponding weights \( \pi_{ij}(x; \Theta) \).

It is worth noting that the form of \( p(y \mid x; \Theta) \) depends on our choice of the kernel function. If we choose a uniform kernel with a range of 1 pixel, it is equivalent to the likelihood for a categorical distribution where each category represents the discrete landmark coordinate. Here we choose a Gaussian kernel to achieve a smoothing effect similar to kernel density estimation.

In this way, we only change the loss function of the neural network for face alignment problem without modifying the heatmap regression based network structure. The goal is to maximize the conditional likelihood without assuming any specific distribution of the target, unlike widely practiced loss function which puts a fixed Gaussian heatmap around the ground truth label as the ground truth heatmap and minimize the L-2 distance between the groundtruth heatmap and the predicted one.

**Loss function.** The loss function is defined as the negative log conditional likelihood. Given training data \( \mathcal{D} = \{x_k, y_k \mid k = 1, 2, \ldots, N\} \), we minimize the loss function to get \( \Theta^* \) as shown in Eq.(2).

\[
\Theta^* = \arg \min_{\Theta} - \sum_{k=1}^{N} \log p(y_k \mid x_k; \Theta)
\]

\[
= \arg \min_{\Theta} - \sum_{k=1}^{N} \log \sum_{i=1}^{m} \sum_{j=1}^{n} K_{\Sigma}(y_k - y_{ij}) \pi_{ij}(x_k; \Theta)
\]

(2)

To demonstrate why the proposed loss function based on kernel density benefits the learning process of face alignment, we compute the gradient of the loss w.r.t. the layer before softmax. To simplify the notation, let \( w_{kij} = K_{\Sigma}(y_k - y_{ij}) \). Denote the layer before softmax for the sample \( k \) as \( f_{kij} \), and the layer after softmax as \( p_{kij} = \text{softmax}(f_{kij}) \). The derivative of the loss contributed by a training sample \( \{x_k, y_k\} \) can be computed as
where \( \frac{\partial \text{Loss}_k}{\partial f_{kij}} = \frac{p_{kij}(w_{kij} - \sum_{a=1}^{m} \sum_{b=1}^{n} w_{kab} p_{kab})}{\sum_{a=1}^{m} \sum_{b=1}^{n} w_{kab} p_{kab}} \) (3)

The proposed target distribution in Eq.(1) is composed of a mixture of Gaussian distributions, thus its covariance matrix can be computed as

\[
\text{Cov}[y | x; \Theta] = \Sigma + \sum_{i=1}^{m} \sum_{j=1}^{n} (y_m - y_{ij})(y_m - y_{ij})^T \pi_{ij}(x; \Theta)
\] (5)

where \( y_m = \sum_{i=1}^{m} \sum_{j=1}^{n} y_{ij} \pi_{ij}(x; \Theta) \).

The uncertainty of the prediction is quantified by the square root of the determinant of the covariance matrix

\[ \sqrt{|\text{Cov}[y | x; \Theta]|} \]

### 3.2. Cascade Probability Propagation

To take advantage of the probability map, we extend the single stage model to multi-stage so that the probability map from the previous stage can be fed into the next stage to progressively improve the landmark estimation accuracy.

Similar to [30], we want to propagate the estimated probability map to the next stage. For each stage, we will have an estimated probability map \( p(y | x; \Theta) \), the raw logits map (before softmax) is concatenated with the down-sampled input image and the feature map with the same size as input to the next stage, as shown in Fig. 1.

The idea is that by propagating the probability map to the next stage, it will guide the network in the next stage implicitly to focus more on the regions in the image with high probability. For example, if the prediction for a certain landmark has high uncertainty, the probability map will have a flat shape, thus encouraging the network at next stage to search in a wider region according to the probability map; if otherwise, the probability map will have a sharp shape and
the network at next stage will try to refine the prediction in the nearby neighborhood.

According to our experiments, the probability map in the first stage is more spread-out, i.e. the prediction is more uncertain, compared to the predictions in later stages. And the prediction error is larger in the first stage than in later stages. This proves the effectiveness of the Kernel Density Network to model output probability distribution as well as the effectiveness of the cascade framework in improving detection accuracy.

Recently, [22] proposed to use the softlabel loss in multiple stages and reducing the fixed variance of the ground truth heatmap stage by stage for a more fine-grained supervision in later stages. One defect of this method is that since it is sensitive to the fixed variance of the ground truth heatmap, it requires careful tuning of this hyperparameter which can be time-consuming for deep neural networks while our method does not require such tuning but automatically learns a more concentrated probability map stage by stage.

4. Experiments

Datasets. We evaluate our methods on 300W [36], Menpo [51], COFW [4], AFLW [24].

300W has 68 landmark annotation. We first train the method on 300W-LP [55] dataset (61225 faces) which is augmented from original 300W dataset for large yaw pose. And then we fine tune on the original trainset (3837 faces). Testing is performed on 300W testset which contains 600 images.

Menpo contains images from AFLW and FDDB with landmark re-annotation following the 68 landmark annotation scheme. It has two subsets, frontal which has 68 landmark annotation for near frontal faces (6679 samples) and profile which has 39 landmark annotation for profile faces (2300 samples). We use the frontal set for cross dataset evaluation.

COFW has 1345 training samples and 507 testing samples, whose facial images are all partially occluded. The dataset is annotated with 29 landmarks. We also perform test on 300W test dataset. The result of softlabel loss [3], we use the same training and testing procedure.

Training procedure: The initial learning rate is $10^{-4}$ for 15 epochs using a minibatch of 10, then dropped to $10^{-5}$ and $10^{-6}$ after every 15 epochs and keep training until convergence. Adam optimizer is used. We apply random augmentations such as random cropping, rotation, flipping, scale noise, color jittering, occlusion, etc.

Testing procedure: We follow the standard testing procedure. The face is cropped using the ground truth bounding box defined in 300W by the extreme locations of the 68 ground truth landmarks. The cropped face is rescaled to $256 \times 256$ before passed to the network. We did not use any other transformation/normalization of the face for fair comparisons.

Overall complexity: The total number of parameters is 23,820,176 $\approx 24M$ in the network with 4 HourGlass modules. With 1 Nvidia RTX 2080 Ti GPU, 1 Xeon CPU, TensorFlow 1.14.0, it takes about 26min to train 1 epoch on 300W-LP dataset and 1.5min on 300W-train dataset. The inference speed is around 10 fps. Our inference is based on the mode of the predicted continuous distribution, obtained by gradient ascent (details in Section 3.1).

4.1. Comparison with existing approaches

We perform test on 300W test dataset. The result of softlabel, KDN-Uniform and KDN-Gaussian are implemented by ourselves using the same structure but different loss functions. To make a fair comparison, they are trained using the same random seed. Result of the softlabel based on our implementation is slightly worse than [3]. The results are shown in Table 1. The CED curves for the 300W test dataset and Menpo Challenge dataset are shown in Fig. 2a and 2b respectively.

We could see from Table 1 that compared to the softlabel loss, our loss function achieves comparable or better performance in terms of NME, AUC and FR and compared to the pseudo NLL computed from softlabel method by normalizing the final heatmap as the probability map, our method gives significantly better NLL. We also compare the results of using a uniform kernel instead of a Gaussian. Using a uniform kernel is equivalent to treating the problem as classi-
Table 1: Prediction results on 300W-test, Menpo-frontal and COFW-68 test (%)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>300W-test</th>
<th>Menpo-frontal</th>
<th>COFW-68 test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NME</td>
<td>AUC</td>
<td>FR</td>
<td>NLL</td>
</tr>
<tr>
<td>TCDCN [52]</td>
<td>4.15</td>
<td>42.1</td>
<td>4.83</td>
<td>-</td>
</tr>
<tr>
<td>CFSS [53]</td>
<td>3.09</td>
<td>56.7</td>
<td>1.83</td>
<td>-</td>
</tr>
<tr>
<td>FAN [3]</td>
<td>2.32</td>
<td>66.5</td>
<td>0.00</td>
<td>-</td>
</tr>
<tr>
<td>SAN [11]</td>
<td>2.86</td>
<td>59.7</td>
<td>1.00</td>
<td>-</td>
</tr>
<tr>
<td>softlabel</td>
<td>2.32</td>
<td>66.6</td>
<td>0.33</td>
<td>4.67</td>
</tr>
<tr>
<td>KDN-Uniform</td>
<td>2.38</td>
<td>65.9</td>
<td>0.50</td>
<td>2.78</td>
</tr>
<tr>
<td>KDN-Gaussian (proposed)</td>
<td>2.21</td>
<td>68.3</td>
<td>0.50</td>
<td>2.93</td>
</tr>
</tbody>
</table>

(a) 300W testset  (b) Menpo-frontal  (c) COFW-68 testset

Figure 2: CED curves of different methods on 300W-test, Menpo-frontal, COFW-68 test

Fication and the target distribution as categorical with the categories representing the different pixel locations. This will result in a very sharp probability map, i.e., an over-confident prediction. And the loss function does not take into account the spatial relationship between different categories.

Fig. 2a shows that our method performs better in some challenging images than the softlabel method. And using a uniform kernel generates slightly larger error compared to using a Gaussian kernel. This is partly because using a uniform kernel introduces quantization error during both training and testing. The quantization error also exists in most recent work adopting the heatmap regression framework, which obtains the landmark coordinate prediction by taking the coordinates of the max value from the output heatmap. But since the heatmap is 4 times smaller than both the width and height of the original input image. This will lead to the downsampling error which makes it difficult to distinguish between the locations of two very close but different landmarks. This can be a big problem for dense landmark schemes. Previous works usually either do not address this issue or address this issue by a heuristic post-processing method such as the implementation provided in [3]. Different from these works, our method constructs a continous mixture of 2D Gaussian distribution from the predicted heatmap. Therefore during testing, we are able to find the mode of the continous distribution even if it lies between two pixels.

4.1.1 Cross-dataset Evaluation

Besides 300W testset, we evaluate the proposed method on Menpo dataset, COFW-68 testset for cross dataset evaluation. The results are shown in Table 1. The method is trained on 300W-LP and fine-tuned on 300W Challenge train set for 68 landmarks. Though the proposed method has similar or marginal improvement on 300W testset and Menpo-frontal dataset, we can see that for cross dataset evaluation on more challenging dataset such as COFW with heavy occlusion, the proposed method shows better performance, especially in terms of NLL.

4.1.2 Probability map visualization

Fig. 3 demonstrates that the proposed method can distinguish between occluded uncertain landmarks and non-occluded landmarks based on predicted heatmap. For occluded landmarks, the predicted heatmap usually has a flatter shape than the non-occluded ones. While the traditional softlabel regression methods can hardly demonstrate the predictive uncertainty in occluded landmarks. Kernel Density Network with a uniform kernel is also able to distinguish occluded landmarks, but it has a sharper shape compared to Gaussian kernel. Similar as in Kernel Density Estimation, Gaussian kernel to some extent smooths the estimated distribution compared to a uniform kernel.

Therefore, our predicted heatmap may be used to detect occlusion without occlusion annotation as supervision, unlike work in [48, 47].

Fig. 4 demonstrates that the proposed method can capture distribution with a more flexible shape. For landmarks lie on the facial contour, the predicted heatmap usually has a shape along the local edge of the face. While the traditional softlabel regression method still predicts a circular shape that represents a standard 2D Gaussian.
Figure 3: Sample heatmaps generated from two methods for occluded landmarks (best viewed in color and magnified). The 1st row is the proposed kernel density method, the 2nd row is the softlabel method. The displayed landmarks are subsets of the 68 points, i.e. the first 3 columns show point 1,5,9,13,17; the last column show point 31,46,37,49,55.

Figure 4: Sample heatmaps generated from two methods with flexible distribution shape (best viewed in color and magnified). The 1st row is the proposed kernel density method, the 2nd row is the softlabel method. The displayed landmarks are subsets of the 68 points, i.e. point 1,5,9,13,17.

4.1.3 Occlusion Dataset

We evaluate the occlusion detection quantitatively on COFW and AFLW-full. For COFW, we report the results on original testset with 29 points annotation and on COFW-68 testset [14] with model trained on 300W train set. Note that for occlusion detection, we are only using the square root of the determinant of covariance computed from the probability map but not any occlusion annotation from dataset or from manual augmentation during training. To compute pseudo variance for softlabel method, we first normalize the heatmap to make the non-negative values sum to one, then treat the normalized heatmap as a probability map to compute variance. KDN-Uniform and KDN-Gaussian generally achieve better precision/recall than softlabel. Since there are other causes of uncertainty besides occlusion, occluded landmarks should have higher uncertainty but not vice versa.

Table 2: Occlusion dataset prediction result (%)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>COFW-29</th>
<th>AFLW-full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>NME</td>
<td>AUC</td>
</tr>
<tr>
<td>SAN [11]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Wing [12]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Softlabel</td>
<td>2.51</td>
<td>64.3</td>
</tr>
<tr>
<td>KDN-Uniform</td>
<td>2.52</td>
<td>64.4</td>
</tr>
<tr>
<td>KDN-Gaussian</td>
<td>2.28</td>
<td>67.8</td>
</tr>
</tbody>
</table>

Table 3: Occlusion detection result (precision/recall %)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>COFW-68</th>
<th>COFW-29</th>
<th>AFLW-full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>NME</td>
<td>AUC</td>
<td>FR</td>
</tr>
<tr>
<td>Softlabel</td>
<td>56/40</td>
<td>61/40</td>
<td>61/40</td>
</tr>
<tr>
<td>KDN-Uniform</td>
<td>70/40</td>
<td>76/40</td>
<td>72/40</td>
</tr>
<tr>
<td>KDN-Gaussian</td>
<td>70/40</td>
<td>75/40</td>
<td>73/40</td>
</tr>
</tbody>
</table>

Table 4: Occluded vs. non-occluded points performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>COFW-68 testset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>NME (%) uncertainty</td>
</tr>
<tr>
<td>Softlabel</td>
<td>2.30 5.99</td>
</tr>
<tr>
<td>KDN-Uniform</td>
<td>2.46 1.25</td>
</tr>
<tr>
<td>KDN-Gaussian</td>
<td>2.34 1.63</td>
</tr>
</tbody>
</table>

4.1.4 Challenging conditions

We evaluate different methods on challenging conditions caused by either low resolution or high noise. We manually add different scales of noise to clean 300W testset and plot the prediction error in NME in Fig. 5a, where we can see that for each method, the prediction error generally increases with noise scale but the proposed method performs best under noisy conditions. In Fig. 5b we show the NME versus the resolution of the input image in pixels.

4.2 Ablation Study

If not specified, ablation study is performed on 300W test set with models trained on 300W-LP and fine-tuned on 300W trainset.

4.2.1 Kernel Density Network

To analyze the effect of the proposed Kernel Density Network, we evaluate the performance of a single stage network in terms of prediction accuracy and uncertainty quantification. Table 5 shows the comparison of results generated from different loss function with a single stage. The proposed loss function is better than the result from softlabel loss.

Table 5: Single stage’s prediction accuracy on 300W testset

<table>
<thead>
<tr>
<th>Method</th>
<th>NME</th>
<th>AUC</th>
<th>FR</th>
<th>NLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softlabel</td>
<td>2.58</td>
<td>62.5</td>
<td>1.00</td>
<td>4.79</td>
</tr>
<tr>
<td>KDN-Uniform</td>
<td>2.57</td>
<td>63.1</td>
<td>1.00</td>
<td>2.95</td>
</tr>
<tr>
<td>KDN-Gaussian</td>
<td>2.52</td>
<td>63.9</td>
<td>0.50</td>
<td>3.01</td>
</tr>
</tbody>
</table>

4.2.2 Multi-stage Cascade

The multi-stage cascade network is trained end-to-end. To analyze the effect of multiple stages, we evaluate the performance of each stage. The NME and average uncertainty at each stage is shown in Fig. 6. From the table we can see that the next stage refines the previous stage’s prediction.
progressively. After each stage, the prediction error reduces and the predicted uncertainty also reduces.

Figure 6: Uncertainty and prediction error at each stage.

4.3. Extension to Other Tasks

Theoretically the proposed method can be widely applied to any regression tasks whose target values are bounded. To demonstrate the generalizability to other tasks, we evaluate the methods on facial action unit intensity estimation.

4.3.1 Facial action unit intensity estimation

We use BP4D dataset and use the metric mean absolute error (MAE) and intra-class correlation (ICC). We divide the dataset into training and testing by different subjects, i.e. training set consists of subjects with odd index and testing set consists of subjects with even index. Results are shown in Table 6. The performance of KDN-Gaussian is not always the best in terms of accuracy, but it gives consistent improvement over KDN-Uniform.

Table 6: Action unit intensity estimation on BP4D dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>ICC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>0.847</td>
<td>0.628</td>
</tr>
<tr>
<td>Gaussian</td>
<td><strong>0.748</strong></td>
<td><strong>0.664</strong></td>
</tr>
<tr>
<td>KDN-Uniform</td>
<td>0.795</td>
<td>0.559</td>
</tr>
<tr>
<td>KDN-Gaussian</td>
<td>0.757</td>
<td>0.588</td>
</tr>
</tbody>
</table>

5. Conclusion

This paper introduced a Kernel Density Deep Neural Network to quantify aleatoric uncertainty in face alignment, and for a more general distribution thus our method is applicable to other regression tasks. Since previous works using fixed variance Gaussian blob heatmap for supervision (softlabel) such as [3] do not quantify different uncertainties of different landmarks, which makes it difficult to apply to real-world problems and tasks that depend on face alignment. To our best knowledge, this is the first work to explicitly address the uncertainty quantification in fully-convolutional neural network based regression problems with a more flexible distribution than Gaussian. We show that uncertainty can be used to detect occlusion without occlusion supervision. Besides, our model provides a principled way of inference using the mode of the predicted continuous distribution to reduce quantization error compared to previous post-processing method such as interpolation [11] or heuristic method [3]. Moreover, in a multi-stage framework, the average predicted uncertainty is reduced stage by stage automatically without manually tuning the variance of the Gaussian blob heatmap in each stage.

We hope this work can benefit the landmark localization community as well as other deep ordinary regression tasks and provide a different perspective in designing the loss function to consider label distribution and aleatoric uncertainty.

Acknowledgment This work is partially supported by Cognitive Immersive Systems Laboratory (CISL), a collaboration between IBM and RPI, and also a center in IBM’s AI Horizon Network.
References


[41] Xuehan Xiong and Fernando De la Torre Frade. Supervised descent method and its applications to face alignment. In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), May 2013.


