Deep Label Distribution Learning With Label Ambiguity

Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, *Member, IEEE,* and Xin Geng, *Member, IEEE*

***Abstract*—Convolutional Neural Networks (ConvNets) have achieved excellent recognition performance in various visual recognition tasks. A large labeled training set is one of the most important factors for its success. However, it is difficult to collect sufficient training images with precise labels in some domains such as apparent age estimation, head pose estimation, multi-label classification and semantic segmentation. Fortunately, there is ambiguous information among labels, which makes these tasks different from traditional classification. Based on this observation, we convert the label of each image into a discrete label distribution, and learn the label distribution by minimizing a Kullback-Leibler divergence between the predicted and ground-truth label distributions using deep ConvNets. The proposed DLDL (Deep Label Distribution Learning) method effectively utilizes the label ambiguity in both feature learning and classifier learning, which help prevent the network from over-fitting even when the training set is small. Experimental results show that the proposed approach produces significantly better results than state-of-the-art methods for age estimation and head pose estimation. At the same time, it also improves recognition performance for multi-label classification and semantic segmentation tasks.**

***Index Terms*—Label distribution, deep learning, age estimation, head pose estimation, semantic segmentation.**

I. Introduction

CONVOLUTIONAL Neural Networks (ConvNets) have achieved state-of-the-art performance on various visual recognition tasks such as image classification [1], object detection [2] and semantic segmentation [3]. The availability of a huge set of training images is one of the most important factors for their success. However, it is difficult to collect sufficient training images with unambiguous labels in domains such as age estimation [4], head pose estimation [5], multi-label classification and semantic segmentation. Therefore, exploiting deep learning methods with limited samples and ambiguous labels has become an attractive yet challenging topic.

Why is it difficult to collect a large and accurately labeled training set? Firstly, it is difficult (even for domain experts) to provide exact labels to some tasks. For example, the pixels close to object boundaries are very difficult to label for annotators in semantic segmentation. In addition, pixel labeling is a time-consuming task that may limit the amount of training samples. Another example is that people's apparent age and head pose is difficult to describe with an accurate number. Secondly, it is very hard to gather complete and sufficient data. For example, it is difficult to build an age dataset covering people from 1 to 85 years old, and ensure that every age in this range has enough associated images. Similar difficulties arise in head pose estimation, where head poses are usually collected at a small set of angles with a 10° or 15° increment. Thus, the publicly available age, head pose and semantic segmentation datasets are small scale compared to those in image classification tasks.

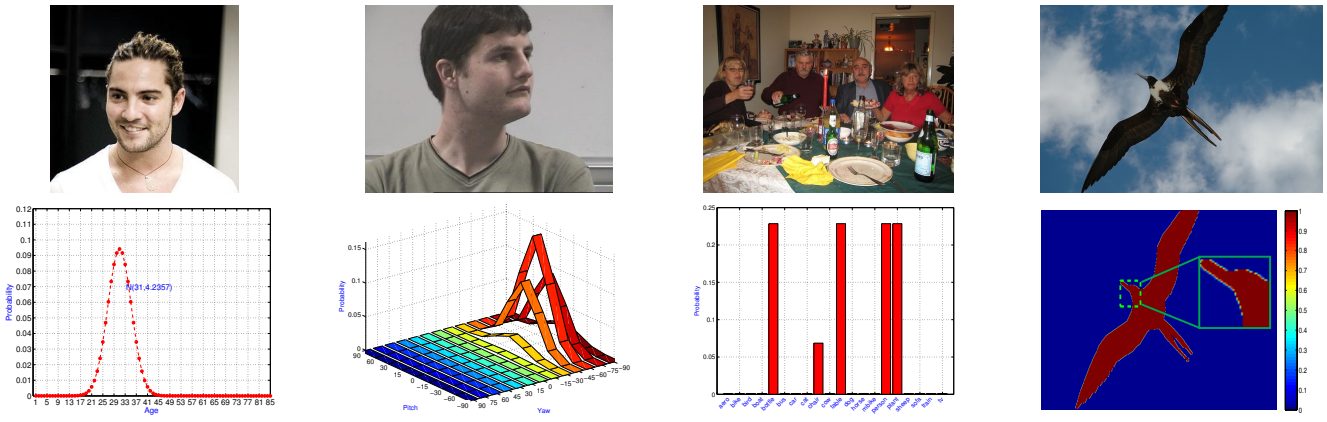
These aforementioned small datasets have a common characteristic, *i.e.*, label ambiguity, which refers to the uncertainty among the ground-truth labels. On one hand, label ambiguity is unavoidable in some applications. We usually predict another person's age in a way like "around 25", which indicates using not only 25, but also neighboring ages to describe the face. And, different people may have different guesses towards the same face. Similar situations also hold for other types of tasks. The labels of pixels at object boundaries are difficult to annotate because of the inherent ambiguity of these pixels in semantic segmentation. On the other hand, label ambiguity can also happen if we are not confident in the labels we provide for an image. In the multi-label classification task, some objects are clearly visible but difficult to recognize. This type of objects are annotated as `Difficult` in the PASCAL Visual Object Classes (VOC) classification challenge [6], *e.g.*, the chair in the third image of the first row in Fig. 1.

There are two main types of labeling methods: single-label recognition (SLR) and multi-label recognition (MLR). SLR assumes one image or pixel has one label and MLR assumes that one image or pixel may be assigned multiple labels. Both SLR and MLR aim to answer the question of which labels can be used to describe an image or pixel, but they can not describe the label ambiguity associated with it. Label ambiguity will help improve recognition performance if it can be reasonably exploited. In order to utilize label correlation (which may be considered as a consequence of label ambiguity in some applications), Geng *et al.* proposed a label distribution learning (LDL) approach for age estimation [4] and head pose estimation [7]. Recently, some improvements of LDL have been proposed. Xing *et al.* proposed two algorithms named LDLogitBoost and AOSO-LDLogitBoost to learn general models to relax the maximum entropy model in traditional

This work was supported in part by the National Natural Science Foundation of China under Grant 61422203, Grant 61622203, and Grant 61232007, in part by the Jiangsu Natural Science Funds for Distinguished Young Scholar under Grant BK20140022, in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization, and in part by the Collaborative Innovation Center of Wireless Communications Technology. (*Corresponding Author: Jianxin Wu.*)

B.-B. Gao, C.-W. Xie and J. Wu are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: gaobb@lamda.nju.edu.cn; xiecw@lamda.nju.edu.cn; wujx@lamda.nju.edu.cn).

C. Xing and X. Geng are with the MOE Key Laboratory of Computer Network and Information Integration, School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: xingchao@seu.edu.cn; xgeng@seu.edu.cn).



(a) Age estimation (b) Head pose estimation (c) Multi-label classification (d) Semantic segmentation

Figure 1. Different label distributions for different recognition tasks. The first row shows four images, with the first two images coming from *ChaLearn 2015* and *Pointing'04* and the last two images coming from the PASCAL *VOC2007* classification task and the PASCAL *VOC2011* segmentation challenge. The second row shows their corresponding label distributions (best viewed in color).

LDL methods [8]. Furthermore, He *et al.* generated age label distributions through weighted linear combination of the input image's label and its context-neighboring samples [9]. However, these methods are suboptimal because they only utilize the correlation of neighboring labels in classifier learning, but not in learning the visual representations.

Deep ConvNets have natural advantages in feature learning. Existing ConvNet frameworks can be viewed as classification and regression models based on different optimization objective functions. In many cases, the softmax loss and ℓ_2 loss are used in deep ConvNet models for classification [10] and regression problems [11], respectively. The softmax loss maximizes the estimated probability of the ground-truth class without considering other classes, and the ℓ_2 loss minimizes the squared difference between the estimated values of the network and the ground-truth. These methods have achieved satisfactory performance in some domains such as image classification, human pose estimation and object detection. However, existing deep learning methods cannot utilize the label ambiguity information. Moreover, a well-known fact is that learning a good ConvNet requires a lot of images.

In order to solve the issues mentioned above, we convert both traditional SLR and MLR problems to *label distribution learning* problems. Every instance is assigned a discrete label distribution $\boldsymbol{y}$ according to its ground-truth. The label distribution can naturally describe the ambiguous information among all possible labels. Through deep label distribution learning, the training instances associated with each class label is significantly increased without actually increase the number of the total training examples. Fig. 1 intuitively shows four examples of label distribution for different recognition tasks. Then, we utilize a deep ConvNet to learn the label distribution in both feature learning and classifier learning. Since we learn label distribution with deep ConvNets, we call our method DLDL: Deep Label Distribution Learning. The benefits of DLDL are summarized as follows:

- DLDL is an end-to-end learning framework which utilizes the label ambiguity in both feature learning and classifier learning;
- DLDL not only achieves more robust performance than existing classification and regression methods, but also effectively relaxes the requirement for large amount of training images, *e.g.*, a training face image with ground-truth label 25 is also useful for predicting faces at age 24 or 26;
- DLDL (only single model without ensemble) achieves better performance than the state-of-the-art methods on age and head pose estimation tasks. DLDL also improves the performance for multi-label classification and semantic segmentation.

The rest of this paper is organized as follows. We first review the related work in Section II. Then, Section III proposes the DLDL framework, including the DLDL problem definition, DLDL theory, label distribution construction and training details. After that, the experiments are reported in Section IV. Finally, Section V presents discussions and the conclusion is given in Section VI.

II. Related Work

In the past two decades, many efforts have been devoted to visual recognition, including at least image classification, object detection, semantic segmentation, and facial attribute (apparent age and head pose) estimation. These works can be divided into two streams. Earlier research was mainly based on hand-crafted features, while more recent ones are usually deep learning methods. In this section, we briefly review these related approaches.

Methods based on hand-crafted features usually include two stages. The first stage is feature extraction. The second stage learns models for recognition, detection or estimation using these features. SVM, random forest [12] and neural networks have commonly been used during the learning stage. In addition, Geng *et al.* proposed the label distribution learning approach to utilize the correlation among adjacent labels, which further improved performance on age estimation [4] and head pose estimation [7].

Although important progresses have been made with these features, the hand-crafted features render them suboptimal for particular tasks such as age or head pose estimation. More recently, learning feature representation has shown great advantages. For example, Lu *et al.* [13] tried to learn cost-sensitive local binary features for age estimation.

Deep learning has substantially improved upon the state-of-the-art in image classification [10], object detection [2], semantic segmentation [3] and many other vision tasks. In many cases, the softmax loss is used in deep models for classification [10]. Besides classification, deep ConvNets have also been trained for regression tasks such as head pose estimation [14] and facial landmark detection [15]. In regression problems, the training procedure usually optimizes a squared ℓ_2 loss function. Satisfactory performance has also been obtained by using Tukey's biweight function in human pose estimation [11]. In terms of model architecture, deep ConvNet models which use deeper architecture and smaller convolution filters (*e.g.*, VGG-Nets [16] and VGG-Face [17]) are very powerful. Nevertheless, these deep learning methods do not make use of the presence of label ambiguity in the training set, and usually require a large amount of training data.

A latest approach, in Inception-v3 [18], is based on label smoothing (LS). Instead of only using the ground-truth label, they utilize a mixture of the ground-truth label and a uniform distribution to regularize the classifier. However, LS is limited to the uniform distribution among labels rather than mining labels' ambiguous information. We believe that label ambiguity is too important to ignore. If we make good use of the ambiguity, we expect the required number of training images for some tasks could be effectively reduced.

In this paper, we focus on how to exploit the label ambiguity in deep ConvNets. Age and head pose estimation from still face images are suitable applications of the proposed research. In addition, we also extend our works to multi-label classification and semantic segmentation.

III. The Proposed DLDL Approach

In this section, we firstly give the definition of the DLDL problem. Then, we present the DLDL theory. Next, we propose the construction methods of label distribution for different recognition tasks. Finally, we briefly introduce the DLDL architecture and training details.

A. The deep label distribution learning problem

Given an input image, we are interested in estimating a category output y (*e.g.*, age or head pose angles). For two input images X^1 and X^2 with ground-truth labels y^1 and y^2, X^1 and X^2 are supposed to be similar to each other if the correlation of y^1 and y^2 is strong, and vice versa. For example, the correlation between faces aged 32 and 33 should be stronger than that between faces aged 32 and 64, in terms of facial details that reflect the age (*e.g.*, skin smoothness). In other words, we expect high correlation among input images with similar outputs. The label distribution learning approach [4], [7] exploited such correlations in the machine learning phase, but used features that are extracted ignoring these correlations. The proposed DLDL approach, however, is an end-to-end deep learning method which utilizes such correlation information in both feature learning and classifier learning. We will also extend DLDL to handle other types of label ambiguity beyond correlation.

To fulfill this goal, instead of outputting a single value y for an input X, DLDL quantizes the range of possible y values into several *labels*. For example, in age estimation, it is reasonable to assume that $0 < y \leq 85$, and it is a common practice to estimate integer values for ages. Thus, we can define the set $L = \{1, 2, \ldots, 85\}$ as the ordered label set for age estimation. The task of DLDL is then to predict a label distribution $\boldsymbol{y} \in \mathbb{R}^{85}$, where y_i is the estimated probability that X should be predicted to be i years old. By estimating an entire label distribution, the deep learning machine is forced to take care of the ambiguity among labels.

Specifically, the input space of our framework is $\mathscr{X} = \mathbb{R}^{h \times w \times d}$, where h, w and d are the height, width, and number of channels of the input image, respectively. DLDL predicts a *label distribution* vector $\boldsymbol{y} \in \mathbb{R}^{|\mathscr{Y}|}$, where $\mathscr{Y} = \{l_1, l_2, \ldots, l_C\}$ is the label set defined for a specific task (*e.g.*, the L above). We assume $\mathscr{Y}$ is complete, *i.e.*, any possible y value has a corresponding member in $\mathscr{Y}$. A training data set with N instances is then denoted as $D = \{(X^1, \boldsymbol{y}^1), \cdots, (X^N, \boldsymbol{y}^N)\}$. We use boldface lowercase letters like $\boldsymbol{y}$ to denote vectors, and the i-th element of $\boldsymbol{y}$ is denoted as y_i. The goal of DLDL is to directly learn a conditional probability mass function $\hat{\boldsymbol{y}} = p(\boldsymbol{y}|X; \boldsymbol{\theta})$ from D, where $\boldsymbol{\theta}$ is the parameters in the framework.

B. Deep label distribution learning

Given an instance X with label distribution $\boldsymbol{y}$, we assume that $\boldsymbol{x} = \phi(X; \boldsymbol{\theta})$ is the activation of the last fully connected layer in a deep ConvNet. We use a softmax function to turn these activations into a probability distribution, that is,

$$\hat{y}_j = \frac{\exp(x_j)}{\sum_t \exp(x_t)} \,. \tag{1}$$

Given a training data set D, the goal of DLDL is to find $\boldsymbol{\theta}$ to generate a distribution $\hat{\boldsymbol{y}}$ that is *similar* to $\boldsymbol{y}$.

There are different criteria to measure the similarity or distance between two distributions. For example, if the Kullback-Leibler (KL) divergence is used as the measurement of the similarity between the ground-truth and predicted label distribution, then the best parameter $\boldsymbol{\theta}^*$ is determined by

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_k y_k \ln \frac{y_k}{\hat{y}_k} = \operatorname*{argmin}_{\boldsymbol{\theta}} -\sum_k y_k \ln \hat{y}_k \,. \tag{2}$$

Thus, we can define the loss function as:

$$T = -\sum_k y_k \ln \hat{y}_k \,. \tag{3}$$

Stochastic gradient descent is used to minimize the objective function Eq. 3. For any k and j,

$$\frac{\partial T}{\partial \hat{y}_k} = -\frac{y_k}{\hat{y}_k} \,, \tag{4}$$

and the derivative of softmax (Eq. 1) is well known, as

$$\frac{\partial \hat{y}_k}{\partial x_j} = \hat{y}_k \left(\delta_{\{k=j\}} - \hat{y}_j \right) , \tag{5}$$

where $\delta_{\{k=j\}}$ is 1 if $k = j$, and 0 otherwise. According to the chain rule, for any fixed j, we have

$$\frac{\partial T}{\partial x_j} = \sum_k \frac{\partial T}{\partial \hat{y}_k} \frac{\partial \hat{y}_k}{\partial x_j} = -y_j + \hat{y}_j \sum_k y_k = -y_j + \hat{y}_j . \tag{6}$$

Thus, the derivative of T with respect to $\boldsymbol{\theta}$ is

$$\frac{\partial T}{\partial \boldsymbol{\theta}} = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} . \tag{7}$$

Once $\boldsymbol{\theta}$ is learned, the label distribution $\hat{\boldsymbol{y}}$ of any new instance X can be generated by a forward run of the network. If the expected class label is a single one, DLDL outputs $l_{i^*} \in \mathscr{Y}$, where

$$i^* = \underset{i}{\operatorname{argmax}} \, \hat{y}_i . \tag{8}$$

Prediction with multiple labels is also allowed, which could be a set $\{l_i | \hat{y}_i > \xi\}$ where $\xi \in [0, 1]$ is a predefined threshold. If the expected output is a real number, DLDL predicts the expectation of $\hat{y}_i$, as

$$\sum_i \hat{y}_i l_i , \tag{9}$$

where $l_i \in \mathscr{Y}$. This indicates that DLDL is suitable for both classification and regression tasks.

C. Label distribution construction

The ground-truth label distribution $\boldsymbol{y}$ is not available in most existing datasets, which must be generated under proper assumptions. A desirable label distribution $\boldsymbol{y} = (y_1, y_2, \ldots, y_C)$ must satisfy some basic principles: (1) $\boldsymbol{y}$ should be a probability distribution. Thus, we have $y_i \in [0, 1]$ and $\sum_{i=1}^{C} y_i = 1$. (2) The probability values y_i should have difference among all possible labels associated with an image. In other words, a less ambiguous category must be assigned high probability and those more ambiguous labels must have low probabilities. In this section, we propose the way to construct label distributions for age estimation, head pose estimation, multi-label classification and semantic segmentation.

For age estimation, we assume that the probabilities should concentrate around the ground-truth age y. Thus, we quantize y to get $\boldsymbol{y}$ using a normal distribution. For example, the apparent age of a face is labeled by hundreds of users. The ground-truth (including a mean μ and a standard deviation σ) is calculated from all the votes. For this problem, we find the range of the target y (*e.g.*, $0 < y \leq 85$), quantize it into a complete and ordered label set $L = \{l_1, l_2, \ldots, l_C\}$, where C is the label set size and $l_i \in \mathbb{R}$ are all possible predictions for y. A label distribution $\boldsymbol{y}$ is then $(y_1, y_2, \ldots, y_C)$, where y_i is the probability that $y = l_i$ (*i.e.*, $y_i = \Pr(y = l_i)$ for $1 \leq i \leq C$). Since we use equal step size in quantizing y, the normal p.d.f. (probability density function) is a natural choice to generate the ground-truth $\boldsymbol{y}$ from μ and σ:

$$y_j = \frac{p(l_j | \mu, \sigma)}{\sum_k p(l_k | \mu, \sigma)} , \tag{10}$$

where $p(l_j | \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(l_j - \mu)^2}{2\sigma^2} \right)$. Fig. 1a shows a face and its corresponding label distribution. For problems where σ is unknown, we will show that a reasonably chosen σ also works well in DLDL.

For head pose estimation, we need to jointly estimate pitch and yaw angles. Thus, learning joint distribution is also necessary in DLDL. Suppose the label set is $L = \{\boldsymbol{l}_{jk} | j = 1, \cdots, n_1, k = 1, \cdots, n_2\}$, where $\boldsymbol{l}_{jk}$ is a pair of values. That is, we want to learn the joint distribution of two variables. Then, the label distribution $\boldsymbol{y}$ can be represented by an $n_1 \times n_2$ matrix, whose (j, k)-th element is y_{jk}. For example, when we use two angles (pitch and yaw) to describe a head pose, $\boldsymbol{l}_{jk}$ is a pair of pitch and yaw angles. Given an instance X with ground-truth mean $\boldsymbol{\mu}$ and covariance matrix Σ, we calculate its label distribution as

$$y_{jk} = \frac{p(\boldsymbol{l}_{jk})}{\sum_j \sum_k p(\boldsymbol{l}_{jk})} , \tag{11}$$

where $p(l_{jk}) = \frac{1}{2\pi |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\boldsymbol{l}_{jk} - \boldsymbol{\mu})^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{l}_{jk} - \boldsymbol{\mu}) \right)$. In the above, we assume $\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$, that is, the covariance matrix is diagonal. Fig. 1b shows a joint label distribution with head pose pitch $= 0°$ and yaw $= 60°$.

For multi-label classification, a multi-label image always contains at least one object of the class of interest. There are usually multiple labels for an image. These labels are grouped into three different levels, including `Positive`, `Negative` and `Difficult` in the PASCAL VOC dataset [6]. A label is `Positive` means an image contains objects from that category, and `Negative` otherwise. `Difficult` indicates that an object is clearly visible but difficult to recognize. Existing multi-label methods often view `Difficult` as `Negative`, which leads to the loss of useful information. It is not reasonable either if we simply treat `Difficult` as `Positive`. Therefore, a nature choice is to use label ambiguity. We define different probabilities for different types of labels, as

$$p_P > p_D > p_N , \tag{12}$$

for `Positive`, `Difficult` and `Negative` labels, respectively. Furthermore, an ℓ_1 normalization is applied to ensure $\sum_{i=1}^{C} y_i = 1$:

$$y_j = \frac{p(l_j)}{\sum_k p(l_k)} , \tag{13}$$

where $p(l_k)$ equals p_P, p_D or p_N if the label l_k is `Positive`, `Difficult` or `Negative`, respectively. The label distribution is shown for a multi-label image in Fig. 1c.

For semantic segmentation, we need to label a pixel as belonging to one class if it is a pixel inside an object of that class, or as the background otherwise. Let y'_{ijk} denote the annotation of the (i, j)-th pixel, where $k = \{0, 1, \ldots, C\}$ (assuming there are C categories and 0 for background). Fully Convolutional Networks (FCN) have been an effective solution to this task. In FCN [3], a ground-truth label l means that $y'_{ijl} = 1$ and $y'_{ijk} = 0$ for all $k \neq l$. However, it is very difficult to specify ground-truth labels for pixels close to object boundaries, because labels of these pixels are inherently

ambiguous. We propose a mechanism to describe the label ambiguity in the boundaries. Considering a Gaussian kernel matrix $f_{K\times K}$, we replace the original label distribution y' with y'', as

$$y''_{ijk} = \sum_{i'=1}^{K}\sum_{j'=1}^{K} f_{i'j'} \times y'_{i'+(i-1)S-P,j'+(j-1)S-P,k}\,. \quad (14)$$

where $f_{i'j'} \geq 0$, $\sum_{i'=1}^{K}\sum_{j'=1}^{K} f_{i'j'} = 1$, K is the kernel size, P and S are padding and stride sizes. In our experiment, we set $K = 5$, $P = 2$ and $S = 1$, and the generated label distribution is

$$y_{ijk} = \frac{y''_{ijk}}{\sum_k y''_{ijk}}\,. \quad (15)$$

Fig. 1d gives the semantic label distribution for a bird image which shows that the ambiguity is encoded in the label distributions.

D. The DLDL architecture and training details

We use a deep ConvNet and a training set D to learn a $\hat{\boldsymbol{y}}$ as the estimation of $\boldsymbol{y}$. The structure of our network is based on popular deep models such as ZF-Net [19] and VGG-Nets [16]. The ZF-Net consists five convolution layers, followed by three fully connected layers. The VGG-Nets architecture includes 16 or 19 layers. We modify the last fully connected layer's output based on the task and replace the original softmax loss function with the KL loss function. In addition, we use the parameter ReLU [20] for ZF-Net. In our network, the input is an order three tensor $X_{h\times w\times d}$ and the output $\hat{\boldsymbol{y}}$ may be a vector (age estimation and multi-label classification), a matrix (head pose estimation) or a tensor (semantic segmentation).

In this paper, we train the deep models in two ways:

Training from scratch. For ZF-Net, the initialization is performed randomly, based on a Gaussian distribution with zero mean and 0.01 standard deviation, and biases are initialized to zero. The coefficient of the parameter ReLU is initialized to 0.25. The dropout is applied to the last two fully connected layers with rate 0.5. The coefficient of weight decay is set to 0.0005. Optimization is done by Stochastic Gradient Descent (SGD) using mini-batches of 128 and the momentum coefficient is 0.9. The initial learning rate is set to 0.01. The total number of epochs is about 20.

Fine-tuning. Three pre-trained models including VGG-Nets (16-layers and 19-layers) and VGG-Face (16-layers) are used to fine-tune for different tasks. We remove these pre-trained models' classification layer and loss layer, and put in our label distribution layer which is initialized by the Gaussian distribution $N(0, 0.01)$ and the KL loss layer. The learning rates of the convolutional layers, the first two fully-connected layers and the label distribution layer are initialized as 0.001, 0.001 and 0.01, respectively. We fine-tune all layers by back propagation through the whole net using mini-batches of 32. The total number of epochs is about 10 for age estimation and 20 for multi-label classification.

IV. Experiments

We evaluate DLDL on four tasks, *i.e.*, age estimation, head pose estimation, multi-label classification and semantic segmentation. Our implementation is based on MatConvNet [21].[1] All our experiments are carried out on a NVIDIA K40 GPU with 12GB of onboard memory.

A. Age estimation

Datasets. Two age estimation datasets are used in our experiments. The first is *Morph* [22], which is one of the largest publicly available age datasets. There are 55,134 face images from more than 13,000 subjects. Ages range from 16 to 77. Since no `TRAIN/TEST` split is provided, 10-fold cross-validation is used for *Morph*.

The second dataset is from the apparent age estimation competition, the first competition track of the ICCV ChaLearn LAP 2015 workshop [23]. Compared with *Morph*, this dataset (*ChaLearn*) consists of images collected in the wild, without any position, illumination or quality restriction. The only condition is that each image contains only one face. The dataset has 4,699 images, and is split into 2,476 training (`TRAIN`), 1,136 validation (`VAL`) and 1,087 testing (`TEST`) images. The apparent age (*i.e.*, how old does this person look like) of each image is labeled by multiple individuals. The age of face images range from 3 to 85. For each image, its mean age and the corresponding standard deviation are given. Since the ground-truth for `TEST` images are not published, we train on the `TRAIN` split and evaluate on the `VAL` split of *ChaLearn* images.

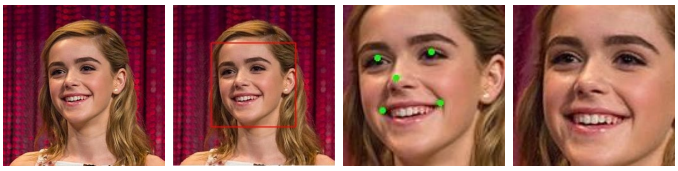
Baselines. To demonstrate the effectiveness of DLDL, we firstly consider two related methods as baselines: ConvNet+LS (KL) and ConvNet+LD (α-div). The former uses label smoothing (LS) [18] as ground-truth and KL divergence as loss function. The latter uses label distribution (LD) as ground-truth and α divergence [24] as loss function, which is

$$T = -2\sum_k (\sqrt{y_k} - \sqrt{\hat{y}_k})^2\,. \quad (16)$$

In addition, we also compare DLDL with the following baseline methods:

- **BFGS-LDL** Geng *et al.* proposed the label distribution learning approach (IIS-LLD) for age and head pose estimation. They used traditional image features. To further improve IIS-LLD, Geng *et al.* [25] proposed a BFGS-LDL algorithm by using the effective quasi-Newton optimization method BFGS.
- **C-ConvNet** Classification ConvNets have obtained very competitive performance in various computer vision tasks. ZF-Net [19] and VGG-Net are popular models which use the softmax loss. We replace the ImageNet-specific 1000-way classification in these modes with the label set $\mathscr{Y}$.
- **R-ConvNet** ConvNets are also successively trained for regression tasks. In R-ConvNet, the ground-truth label y (age and pose angle) is projected into the range $[-1, 1]$

[1] http://www.vlfeat.org/matconvnet/



(a) Input (b) Detection (c) Facial points (d) Alignment

Figure 2. The face image pre-processing pipeline.

by the mapping $\frac{2(y-\min)}{\max-\min}-1$, where max and min are the maximum and minimum values in the training label set. During prediction, the R-ConvNet regression result is reverse mapped to get $\hat{y}$. To speed up convergence, the last fully connected layer is followed a hyperbolic tangent activation function $f(x)=\tanh(x)$, which maps $[-\infty,+\infty]$ to $[-1,+1]$ [14]. The squared ℓ_2, ℓ_1 and ϵ-ins loss functions are used in R-ConvNet.

Implementation details. We use the same preprocessing pipeline for all compared methods, including face detection, facial key points detection and face alignment, as shown in Fig 2. We employ the DPM model [26] to detect the main facial region. Then, the detected face is fed into cascaded convolution networks [15] to get the five facial key points, including the left/right eye centers, nose tip and left/right mouth corners. Finally, based on these facial points, we align the face to the upright pose. Data augmentation are only applied to the training images for *ChaLearn*. For one color input training image, we generate its gray-scale version, and left-right flip both color and gray-scale versions. Thus, every training image turns into 4 images.

We define $\mathscr{Y}=\{1,2,\ldots,85\}$ for both datasets. The label distribution of each image is generated using Eq. 10. The mean μ is provided in both *Morph* and *ChaLearn*. The standard deviation σ, however, is provided in *ChaLearn* but not in *Morph*. We simply set $\sigma=2$ in *Morph*. Experiments for different methods are conducted under the same data splits.

Evaluation criteria. Mean Absolute Error (MAE) and Cumulative Score (CS) are used to evaluate the performance of age estimation. MAE is the average difference between the predicted and the real age:

$$MAE=\frac{1}{N}\sum_{n=1}^{N}|\hat{l}_n-l_n|, \tag{17}$$

where $\hat{l}_n$ and l_n are the estimated and ground-truth age of the n-th testing image, respectively. CS is defined as the accuracy rate of *correct estimation*:

$$CS_g=\frac{C_g}{N}\times 100\%, \tag{18}$$

where C_g is the number of *correct estimation*, *i.e.*, testing images that satisfy $|\hat{l}_n-l_n|\leq g$. In our experiment, $g\in\{1,2,\ldots,30\}$. In addition, a special measurement (named ϵ-error) is defined by the *ChaLearn* competition, computed as

$$\epsilon=\frac{1}{N}\sum_{n=1}^{N}\left(1-\exp\left(-\frac{(\hat{l}_n-l_n)^2}{2\sigma_n^2}\right)\right). \tag{19}$$

Table I
COMPARISONS OF DIFFERENT METHODS FOR AGE ESTIMATION.

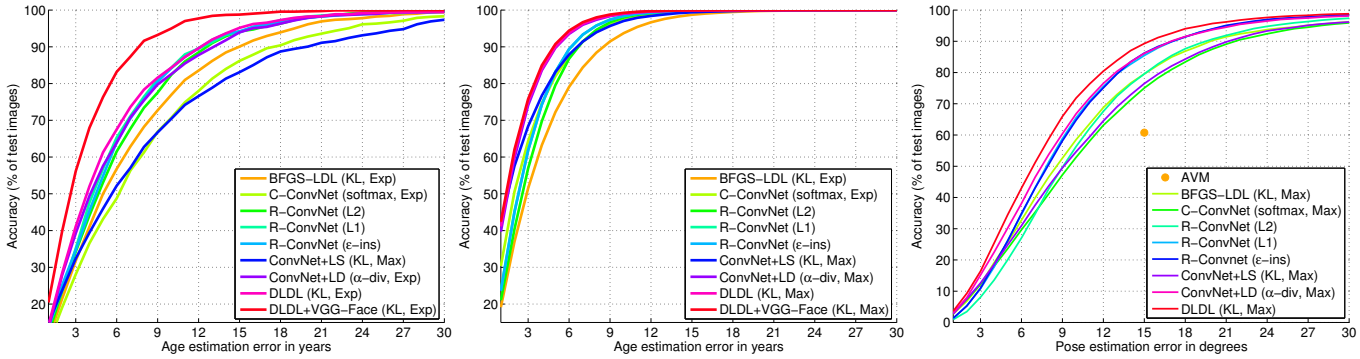
Description	Morph MAE	ChaLearn MAE	ChaLearn ϵ-error
IIS-LDL [4]	5.67±0.15	-	-
CPNN [4]	4.87±0.31	-	-
ST+CSHOR [27][1]	3.82	-	-
M-S ConvNets [28]	3.63	-	-
ConvNets [29][1]	3.31	-	-
VGG (softmax, Exp) [30][3]	-	6.08	0.51
VGG (softmax, Exp) [30][2,3]	-	**3.22**	**0.28**
VGG (softmax, Exp) [31][2,3]	2.68	3.25	**0.28**
BFGS-LDL (KL, Max)	3.94±0.05	7.81	0.57
BFGS-LDL (KL, Exp)	3.85±0.05	6.79	0.53
C-ConvNet (softmax, Max)	3.02±0.05	9.48	0.63
C-ConvNet (softmax, Exp)	2.86±0.05	7.95	0.58
R-ConvNet (ℓ_2)	3.17±0.04	5.94	0.50
R-ConvNet (ℓ_1)	2.88±0.03	5.62	0.47
R-ConvNet (ϵ-ins)	2.89±0.04	5.71	0.48
ConvNet+LS (KL, Max)	2.96±0.13	8.64	0.59
ConvNet+LS (KL, Exp)	5.02±0.13	11.58	0.77
ConvNet+LD (α-div, Max)	2.57±0.04	5.95	0.47
ConvNet+LD (α-div, Exp)	2.57±0.04	5.69	0.46
DLDL (KL, Max)	2.51±0.03	5.49	0.44
DLDL (KL, Exp)	2.52±0.03	5.34	0.44
DLDL+VGG-Face (KL, Max)[3]	**2.42±0.01**	3.62	0.32
DLDL+VGG-Face (KL, Exp)[3]	2.43±0.01	3.51	0.31

[1]Used 80% of Morph images for training and 20% for evaluation;
[2]Used additional external face images (*i.e.*, IMDB-WIKI);
[3]Used pre-trained model (*i.e.*, VGG-Nets or VGG-Face).

Results. Table I lists results on both datasets. The upper part shows results in the literature. The middle part shows the baseline results. The lower part shows the results of the proposed approach. The first term in the parenthesis behind each method is the loss function corresponding to the method. `Max` or `Exp` represent predicting according to Eq. 8 or 9, respectively. Since cross-validation is used in *Morph*, we also provide its standard deviations.

From Table I, we can see that DLDL consistently outperforms baselines and other published methods. The difference between DLDL (KL, Max) and its competitor C-ConvNet (softmax, Max) is 0.51 on *Morph*. This gap is more than 6 times the sum of their standard deviations (0.03+0.05), showing statistically significant differences. The advantage of DLDL over R-ConvNet, C-ConvNet and ConvNet+LS suggests that learning label distribution is advantageous in deep end-to-end models. DLDL has much better results than BFGS-LDL, which shows that the learned deep features are more powerful than manually designed ones. Compared to ConvNet+LD (α-div), DLDL (KL) achieves lower MAE on both datasets. It indicates that KL-divergence is better than α-divergence for measuring the similarity of two distributions in this context.

We find that C-ConvNet and R-ConvNet are not stable. The R-ConvNet (ℓ_1) method, although being the second best method for *ChaLearn*, is inferior to C-ConvNet (softmax, Exp) for *Morph*. In addition, we also find that Eq. 9 is better than Eq. 8 in many cases, which suggests that Eq. 9 is more suitable than Eq. 8 for age estimation.



(a) ChaLearn (b) Morph (c) AFLW

Figure 3. Comparisons of CS curves on the *ChaLearn*, *Morph* and *AFLW* validation sets. Note that the CS cures are plotted using better estimation based on Table I for those methods involving Max (Eq. 8) and Exp (Eq. 9) (higher is better, best viewed in color).

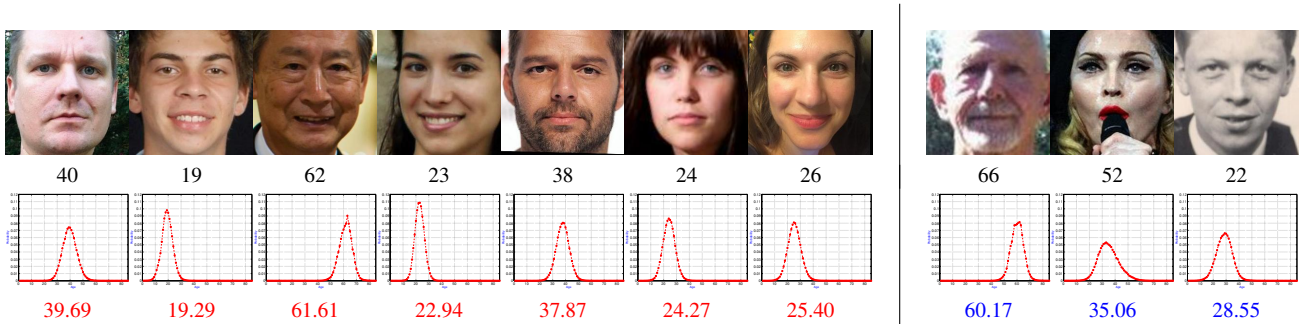


Figure 4. Examples of face images and DLDL results. The first row shows ten cropped and aligned faces from the apparent age estimation challenge and their corresponding ground-truth apparent ages. The second row shows their predicted label distributions and predicted ages. The left seven columns show good age estimations and the right three columns are failure cases.

Fine-tuning DLDL. Instead of training DLDL from scratch, we also fine-tune the network of VGG-Face [17]. On the small scale *ChaLearn* dataset, the MAE of DLDL is reduced from 5.34 to 3.51, yielding a significant improvement. The ϵ-error of DLDL is reduced from 0.44 to 0.31, which is close to the best competition result 0.28 [30] on the validation set. In [31], external training images (260,282 additional external training images with real age annotation) were used. DLDL only uses the *ChaLearn* dataset's 2,476 training images and is the best among *ChaLearn* teams that do not use external data [23]. In the competition, the best external-data-free ϵ-error is 0.48, which is worse than DLDL's. However, the idea in [31] to use external data is useful for further reducing DLDL's estimation error.

Fig. 3a and Fig. 3b show the CS curves on *ChaLearn* and *Morph* datasets. At every error level, our DLDL fine-tuned VGG-Face always achieves the best accuracy among all methods. It is noteworthy that the CS curves of DLDL (KL, Max) and ConvNet (α-div, Max) are very close to that of the DLDL+VGG-Face (KL, Max) on *Morph* even without lots of external data and very deep model. This observation supports the idea that using DLDL can achieve competitive performance even with limited training samples.

In Fig. 4, we show some examples of face images from the *ChaLearn* validation set and predicted label distributions by DLDL (KL, Exp). In many cases, our solution is able to accurately predict the apparent age of faces. Failures may come from two causes. The first is the failure to detect or align the face. The second is some extreme conditions of face images such as occlusion, low resolution, heavy makeup and old photos.

B. Head pose estimation

Datasets. We use three datasets in head pose estimation: *Pointing'04* [32], *BJUT-3D* [33] and Annotated Facial Landmarks in the Wild (*AFLW*) [34]. In them, head pose is determined by two angles: pitch and yaw. *Pointing'04* discretizes the pitch into 9 angles $\{0°, \pm15°, \pm30°, \pm60°, \pm90°\}$ and the yaw into 13 angles $\{0°, \pm15°, \pm30°, \pm45°, \pm60°, \pm75°, \pm90°\}$. When the pitch angel is $+90°$ or $-90°$, the yaw angle is always set to $0°$. Thus, there are 93 poses in total. The head images are taken from 15 different human subjects in two different time periods, resulting in $15 \times 2 \times 93 = 2,790$ images.

BJUT-3D contains 500 3D faces (250 male and 250 female people), acquired by a CyberWare Laser Scanner in an engineered environment. 9 pitch angles $\{0°, \pm10°, \pm20°, \pm30°, \pm40°\}$ and 13 yaw angles $\{0°, \pm10°, \pm20°, \pm30°, \pm40°, \pm50°, \pm60°\}$ are used. There are in total 93 poses in this dataset, similar to that in *Pointing'04*. Therefore, $500 \times 93 = 46,500$ face images are obtained.

Unlike *Pointing'04* and *BJUT-3D*, the *AFLW* is a real-world face database. Head pose is coarsely obtained by fitting a mean 3D face with the POSIT algorithm [35]. The dataset contains

Table II
COMPARISONS OF DIFFERENT METHODS FOR HEAD POSE ESTIMATION ON THE *Pointing'04* DATASET.

Methods	Description	MAE (lower is better)			Acc (higher is better)		
		Pitch	Yaw	Pitch+Yaw	Pitch	Yaw	Pitch+Yaw
	LDL-wJ [7]	2.69±0.15	4.24±0.17	6.45±0.29	86.24±0.97	73.30±1.36	64.27±1.82
Baselines	BFGS-LDL (KL)	1.99±0.19	4.00±0.20	5.68±0.13	88.78±0.11	74.37±0.13	66.42±0.11
	C-ConvNet (softmax)	5.28±0.65	6.02±0.44	10.56±0.74	73.15±2.74	62.90±1.81	42.97±1.67
	R-ConvNet (ℓ_2)	6.11±0.33	6.61±0.17	10.13±0.26	-	-	-
	R-ConvNet (ℓ_1)	5.94±0.71	5.90±0.39	9.43±0.79	-	-	-
	R-ConvNet (ϵ-ins)	5.77±0.45	6.66±0.19	9.04±0.40	-	-	-
	ConvNet+LS (KL)	5.23±0.39	5.87±0.53	10.42±0.66	72.62±1.01	62.90±2.76	41.83±2.20
	ConvNet+LD (α-div)	1.94±0.20	3.68±0.16	5.34±0.17	90.00±0.77	76.27±0.82	69.00±0.89
Ours	DLDL (KL)	**1.69±0.32**	**3.16±0.07**	**4.64±0.24**	**91.65±1.13**	**79.57±0.57**	**73.15±0.72**

Table III
COMPARISONS OF DIFFERENT METHODS FOR HEAD POSE ESTIMATION ON THE *BJUT-3D* DATASET.

Methods	Description	MAE (lower is better)			Acc (higher is better)		
		Pitch	Yaw	Pitch+Yaw	Pitch	Yaw	Pitch+Yaw
Baselines	BFGS-LDL (KL)	0.19±0.02	0.33±0.04	0.51±0.05	98.15±0.19	96.69±0.38	94.95±0.54
	C-ConvNet (Softmax)	0.06±0.01	0.09±0.02	0.14±0.03	99.45±0.09	99.16±0.16	98.64±0.23
	R-ConvNet (ℓ_2)	1.83±0.01	2.17±0.03	3.15±0.03	-	-	-
	R-ConvNet (ℓ_1)	1.25±0.06	1.37±0.09	2.11±0.09	-	-	-
	R-ConvNet (ϵ-ins)	1.21±0.07	1.42±0.07	2.09±0.10	-	-	-
	ConvNet+LS (KL)	0.05±0.01	0.08±0.01	0.12±0.01	99.55±0.06	99.28±0.08	98.86±0.10
	ConvNet+LD (α-div)	0.07±0.01	0.12±0.02	0.19±0.02	99.31±0.04	98.82±0.20	98.15±0.21
Ours	DLDL (KL)	**0.02±0.01**	**0.07±0.01**	**0.09±0.01**	**99.81±0.04**	**99.27±0.08**	**99.09±0.09**

about 24k faces in real-world images. We select 23,409 faces to ensure pitch and yaw angles within $[-90°, 90°]$.

Implementation details. The head region is provided by bounding box annotations in *Pointing'04* and *AFLW*. The *BJUT-3D* does not contain background regions. Therefore, we will not perform any preprocessing.

In DLDL, we set $\sigma = 15°$ in *Pointing'04* and $\sigma = 5°$ in *BJUT-3D* for constructing label distributions. For *AFLW*, ground-truth of head pose angles are given as real numbers. Ground-truth (pitch and yaw) angles are divided from $-90°$ to $+90°$ in steps of $3°$, so we get $61 \times 61 = 3,721$ (pitch, yaw) pair category labels. We set $\sigma = 3°$ for *AFLW*. Since the discrete Jeffrey's divergence is used in LDL [7], we implement BFGS-LDL with the Kullback-Leibler divergence. All experiments are performed under the same setting, including data splits, input size and network architecture.

To validate the effectiveness of DLDL for head pose estimation, we use the same baselines as age estimation. Our experiments show that Eq. 9 has lower accuracy than Eq. 8. Hence, we use Eq. 8 in this section.

Evaluation criteria. Three types of prediction values are evaluated: pitch, yaw, and pitch+yaw, where pitch+yaw jointly estimates the pitch and yaw angles. Two different measurements are used, which is MAE (Eq. 17) and classification accuracy (`Acc`). When we treat different poses as different classes, `Acc` measures the pose class classification accuracy. In particular, the MAE of pitch+yaw is calculated as the Euclidean distance between the predicted (pitch, yaw) pair and the ground-truth pair; the `Acc` of pitch+yaw is calculated by regarding each (pitch, yaw) pair as a class. For R-ConvNet, we only report its MAE but not `Acc`, because its predicted value are continuous real numbers. All methods are tested with 5-fold cross validation for *Pointing'04* and *BJUT-3D* following [7]. For *AFLW*, 15,561 face images are randomly chosen for training, and the remaining 7,848 for evaluation. The setup is similar to the recent literature [36] (14,000 images for training and the rest 7,041 images for testing).

Table IV
MAE AND ACC (% OF IMAGES WITH $\pm 15°$ ERROR) FOR DIFFERENT METHODS ON THE *AFLW* DATASET.

Description	MAE (lower is better)			Acc (higher is better)		
	Pitch	Yaw	Pitch+Yaw	Pitch	Yaw	Pitch+Yaw
AVM [36]	-	16.75	-	-	60.75	-
BFGS-LDL (KL)	7.21	8.72	12.69	90.62	86.81	79.80
C-ConvNet (softmax)	7.87	9.34	13.65	87.75	83.79	75.04
R-ConvNet (ℓ_2)	6.57	8.44	11.88	92.84	84.76	79.56
R-ConvNet (ℓ_1)	6.01	7.07	10.34	94.60	89.62	85.45
R-ConvNet (ϵ-ins)	5.96	7.13	10.35	94.94	90.00	86.21
ConvNet+LS (KL)	7.69	9.10	13.33	88.34	85.00	76.47
ConvNet+LD (α-div)	6.55	7.02	10.77	92.80	91.88	86.14
DLDL (KL)	**5.75**	**6.60**	**9.78**	**95.41**	**92.89**	**89.27**

Results. Tables II, III and IV show results on *Pointing'04*, *BJUT-3D* and *AFLW*, respectively. *Pointing'04* is small scale with only 2,790 images. We observe that BFGS-LDL (with hand-crafted features) has much lower MAE and much higher accuracy than deep learning methods C-ConvNet, R-ConvNet and ConvNet+LS. One reasonable conjecture is that C-ConvNet, R-ConvNet and ConvNet+LS are not well-learned with only small number of training images. DLDL, however, successfully learns the head pose. For example, its accuracy for pitch+yaw is 73.15% (and C-ConvNet is only 42.97%). That is, DLDL is able to perform deep learning with few training images, while C-ConvNet R-ConvNet and ConvNet+LS have failed for this task.

On *BJUT-3D* and *AFLW* which have enough training data, we observe that many deep learning methods show higher

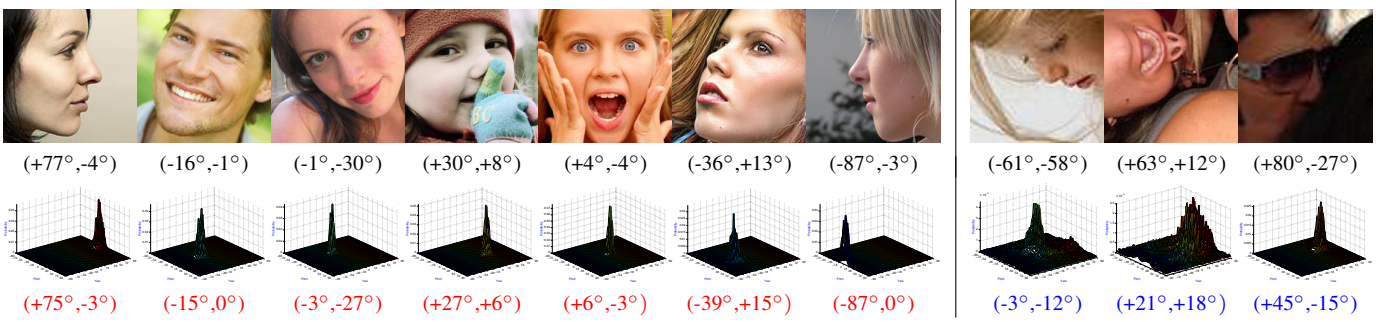


Figure 5. Examples of face images and DLDL results. The first row shows ten cropped faces from the *AFLW* dataset and their corresponding ground-truth labels (yaw angle, pitch angle). The second row shows their predicted label distributions and predicted head poses. The left seven columns are the good examples and the right three columns are the failure cases.

performance than BFGS-LDL. DLDL achieves the best performance: it has much lower MAE and higher accuracy than other methods. Another observation is also worth mentioning. Although R-ConvNet is better than C-ConvNet when label is dense such as age estimation and head pose estimation on *AFLW*, it is obviously worse than C-ConvNet on *BJUT-3D* and *pointing'04* for head pose estimation which have sparse labels. In other words, the performance of C-ConvNet and R-ConvNet are not very robust, while the proposed method consistently achieves excellent performance.

Fig. 3c shows the pitch+yaw CS curves on the *AFLW* dataset. There is an obvious gap between DLDL and baseline methods at every error level. Fig. 5 shows the predicted label distributions for different head poses on the *AFLW* testing set using the DLDL model. Our approach can estimate head pose with low errors but may fail under some extreme conditions. It is noteworthy that DLDL may produce more incorrect estimations when both yaw and pitch are large (*e.g.*, $\pm 90°$). The reason might be that there are much fewer training examples for large angles than for other angles.

C. Multi-label classification

Datasets. We evaluate our approach for multi-label classification on the PASCAL VOC dataset [6]: PASCAL *VOC2007* and *VOC2012*. There are 9,963 and 22,531 images in them, respectively. Each image is annotated with one or several labels, corresponding to 20 object categories. These images are divided into three subsets including `TRAIN`, `VAL` and `TEST` sets. We train on the `TRAINVAL` set and evaluate on the `TEST` set. The evaluation metric is average precision (AP) and mean average precision (mAP), complying with the PASCAL challenge protocols.

We denote our methods as Images-Fine-tuning-DLDL (IF-DLDL) and Proposals-Fine-tuning-DLDL (PF-DLDL) when ConvNets are fine-tuned by images and proposals of images, respectively. Details of these two variants are explained later in this section. We compare the proposed approaches with the following methods:

- **VGG+SVM [16]**. This method densely extracted 4,096 dimensional ConvNet features at the penultimate layer of VGG-Nets pre-trained on ImageNet. These features from different scales (smallest image side $Q \in \{256, 384, 512, 640, 768\}$) were aggregated by average pooling. Then, these averaged features from two networks ("Net-D" containing 16 layers and "Net-E" containing 19 layers) were further fused by stacking. Finally, [16] ℓ_2 normalized the resulting image features and used these features to train a linear SVM classifier for multi-label classification.
- **HCP [37]**. HCP proposed to solve the multi-label object recognition task by extracting object proposals from the images. The method used image label and square loss to fine-tune a pre-trained ConvNet. Then, BING [38] or EdgeBoxes [39] was used to extract object proposals, which were used to fine-tune the ConvNet again. Finally, scores of these proposals were max-pooled to obtain the prediction.
- **Fev+Lv [40]**. This approach transformed the multi-label object recognition problem into a multi-class multi-instance learning problem. Two views (label view and feature view) were extracted for each proposal of images. Then, these two views were encoded by a Fisher vector for each image.
- **IF-VGG-ℓ_2** and **IF-VGG-KL**. We fine-tune the VGG-Nets with square loss and multi-label cross-entropy loss [41] and use them as our IF-DLDL's baselines. They are trained using the same setting.

Table V
SINGLE MODEL CLASSIFICATION MAP (IN %) ON *VOC2007* (`TRAINVAL`/`TEST`). THE * SIGN INDICATES GROUND-TRUTH BOUNDING BOX INFORMATION WAS USED DURING TRAINING.

Methods	Description	Net-D Max	Net-D Avg	Net-E Max	Net-E Avg
	Fev+Lv-20-VD* [40]	90.6	-	-	-
	HCP-VGG [42]	90.9	-	-	-
Baselines	VGG+SVM [16]	89.3	-	89.3	-
	IF-VGG-ℓ_2	89.8	89.5	89.7	89.8
	IF-VGG-KL	90.0	90.3	90.3	90.2
Ours	IF-DLDL	90.1	90.5	90.6	90.7
	PF-DLDL	**92.3**	**92.1**	**92.5**	**92.2**

Implementation details. According to the ground-truth labels, we set different probabilities for all possible labels on PASCAL VOC dataset. In our experiments, $p_P = 1$, $p_D = 0.3$, $p_N = 0$. Finally, similar to label smoothing, a

Table VI
COMPARISONS OF THE CLASSIFICATION RESULTS (IN %) OF STATE-OF-THE-ART APPROACHES ON *VOC2007* (TRAINVAL/TEST). * INDICATES METHODS USING GROUND-TRUTH BOUNDING BOX INFORMATION FOR TRAINING.

Methods	Description	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
	AGS* [46]	82.2	83.0	58.4	76.1	56.4	77.5	88.8	69.1	62.2	61.8	64.2	51.3	85.4	80.2	91.1	48.1	61.7	67.7	86.3	70.9	71.1
	AMM* [47]	84.5	81.5	65.0	71.4	52.2	76.2	87.2	68.5	63.8	55.8	65.8	55.6	84.8	77.0	91.1	55.2	60.0	69.7	83.6	77.0	71.3
	HCP-2000C [37]	96.0	92.1	93.7	93.4	58.7	84.0	93.4	92.0	62.8	89.1	76.3	91.4	95.0	87.8	93.1	69.9	90.3	68.0	96.8	80.6	85.2
	Fev+Lv-20-VD* [40]	97.9	97.0	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	**97.7**	95.9	**98.6**	77.6	88.7	78.0	98.3	89.0	90.6
	HCP-VGG [42]	98.6	97.1	98.0	95.6	75.3	94.7	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
Baselines	VGG+SVM [16]	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	87.8	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
Baselines	IF-VGG-ℓ_2	98.9	95.7	97.3	95.5	65.0	92.8	93.7	97.1	74.2	90.8	87.0	97.1	97.1	93.8	97.0	70.8	94.3	77.8	98.0	86.4	90.0
Baselines	IF-VGG-KL	99.1	95.5	97.4	94.9	68.1	92.7	94.3	97.0	75.7	90.3	89.0	97.0	97.6	94.6	97.2	76.3	93.8	80.1	98.2	87.9	90.8
Ours	IF-DLDL	99.1	95.8	97.4	95.3	69.2	93.3	94.5	96.6	76.1	90.4	**89.0**	97.1	**97.7**	94.5	97.7	76.1	93.6	81.9	98.2	89.1	91.1
Ours	PF-DLDL	**99.3**	**97.6**	**98.3**	**97.0**	**79.0**	**95.7**	**97.0**	**97.9**	**81.8**	**93.3**	88.2	**98.1**	96.9	**96.5**	98.4	**84.8**	**94.9**	**82.7**	**98.5**	**92.8**	**93.4**

Table VII
COMPARISONS OF THE CLASSIFICATION RESULTS (IN %) OF STATE-OF-THE-ART APPROACHES ON *VOC2012* (TRAINVAL/TEST). * INDICATES METHODS USING GROUND-TRUTH BOUNDING BOX INFORMATION FOR TRAINING.

Methods	Description	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
	NUS-PSL*[46]	97.3	84.2	80.8	85.3	60.8	89.9	86.8	89.3	75.4	77.8	75.1	83.0	87.5	90.1	95.0	57.8	79.2	73.4	94.5	80.7	82.2
	PRE-1512*[48]	94.6	82.9	88.2	84.1	60.3	89.0	84.4	90.7	72.1	86.8	69.0	92.1	93.4	88.6	96.1	64.3	86.6	62.3	91.1	79.8	82.8
	HCP-2000C [37]	97.5	84.3	93.0	89.4	62.5	90.2	84.6	94.8	69.7	90.2	74.1	93.4	93.7	88.8	93.3	59.7	90.3	61.8	94.4	78.0	84.2
	Fev+Lv-20-VD* [40]	98.4	92.8	93.4	90.7	74.9	93.2	90.2	96.1	78.2	89.8	80.6	95.7	96.1	95.3	97.5	73.1	91.2	75.4	97.0	88.2	89.4
	HCP-VGG [42]	99.1	92.8	97.4	94.4	79.9	93.6	89.8	**98.2**	78.2	94.9	79.8	97.8	97.0	93.8	96.4	74.3	94.7	71.9	96.7	88.6	90.5
Baselines	VGG+SVM [16]	99.0	89.1	96.0	94.1	74.1	92.2	85.3	97.9	79.9	92.0	83.7	97.5	96.5	94.7	97.1	63.7	93.6	75.2	97.4	87.8	89.3
Baselines	IF-VGG-ℓ_2	98.9	88.4	96.7	93.4	70.7	92.3	85.8	97.7	77.3	94.2	81.2	97.4	96.8	93.7	96.7	62.2	94.1	70.7	96.9	85.8	88.6
Baselines	IF-VGG-KL	99.0	89.9	96.6	93.7	74.0	93.2	87.3	97.5	78.5	94.7	83.1	97.1	96.9	94.0	96.6	66.9	94.5	75.9	97.4	87.7	89.7
Ours	IF-DLDL	99.0	89.7	96.6	94.1	74.8	93.1	87.8	97.6	79.3	94.3	83.4	97.2	96.9	94.0	97.3	67.8	94.2	76.5	97.4	87.8	89.9
Ours	PF-DLDL	**99.5**	**94.1**	**97.9**	**95.9**	**81.0**	**94.8**	**93.1**	**98.2**	**82.4**	**96.1**	**84.0**	**98.0**	**97.8**	**95.7**	**97.7**	**78.9**	**95.5**	**78.0**	**97.8**	**92.2**	**92.4**

uniform distribution $u_i = \epsilon/20$ is added to $\boldsymbol{y}$, where $\epsilon = 0.01$.

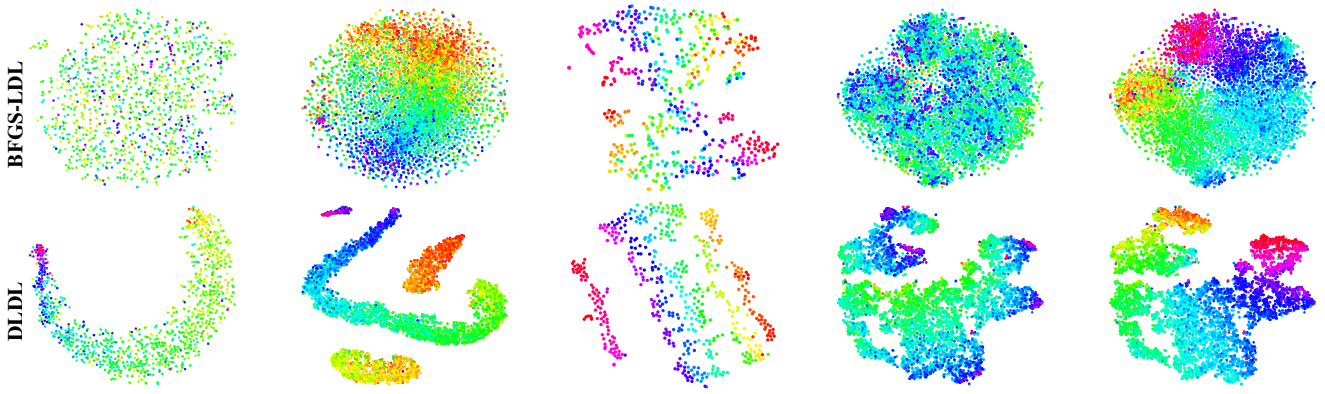
IF-DLDL. Following [16], each training image is individually rescaled by randomly sampling in the range [256, 512]. We randomly crop 256×256 patches from these resized images. We also adjust the pooling kernel in the `pool5` layer from 3×3 to 4×4. Max-pooling and Avg-pooling are used at `pool5` to train two ConvNets. We obtain four ConvNet models thought fine-tuning "Net-D" and "Net-E". At the prediction stage, the smaller side of each image is scaled to a fixed length $Q \in \{256, 320, 384, 448, 512\}$. Each scaled image is fed to the fine-tuned ConvNets to obtain the 20-dim probability outputs. These probability outputs from different scales and different models are averaged to form the final prediction.

PF-DLDL. Following [42], we further fine-tune IF-DLDL models with proposals of images to boost performance. For each training image, we employ EdgeBoxes [39] to produce a set of proposal bounding boxes which are grouped into m clusters by the normalized cut algorithm [43]. For each cluster, the top k proposals with higher predictive scores generated by EdgeBoxes are resized into square shapes (*i.e.*, 256×256). As a result, we can obtain mk proposals for an image. Finally, these mk resized proposals are fed into a fine-tuned IF-DLDL model to obtain prediction scores and these scores are fused by max-pooling to form the prediction distribution of the image. This process can be learned by using an end-to-end way. In our implementation, we set $m = 15, k = 1$ and $m = 15, k = 30$ at the training and the prediction stage, respectively. Similar to IF-DLDL, we also average fuse prediction scores of different models to generate the final prediction.

Results. In Table V, we compare single model results (average AP of all classes) on *VOC2007*. Our PF-DLDL defeats all the other methods. Compared with Fev+Lv [40], 1.7% improvement can be achieved by PF-DLDL even without using the bounding box annotation. Compared with HCP-VGG [42], our PF-DLDL can achieve 92.3% mAP, which is significantly higher than their 90.9%. This further indicates that it is very important to learn a label distribution.

Table VI and VII report details of all experimental results on *VOC2007* and *VOC2012*, respectively. It can be seen that IF-DLDL outperforms IF-VGG-ℓ_2 by 1.1% for *VOC2007* and 1.3% for *VOC2012*, which indicates that the KL loss function is more suitable than ℓ_2 loss for measuring the similarity of two label distributions. Furthermore, IF-DLDL improves IF-VGG-KL for about 0.2–0.3 points in mAP, which suggests that learning a label distribution is beneficial. More importantly, PF-DLDL can achieve 93.4% for *VOC2007* and 92.4% for *VOC2012* in mAP when we average fuse output scores of four PF-DLDL models.

Our framework shows good performance especially for scene categories such as "chair", 'table" and "sofa". Although PF-DLDL significantly outperforms IF-DLDL in mAP, PF-DLDL has higher computational cost than IF-DLDL on both training and testing stages. Since IF-DLDL does not need region proposals or bounding box information, it may be effectively and efficiently implemented for practical multi-label application such as multi-label image retrieval [44]. It is also possible that by adopting new techniques (such as the region proposal method using gated unit in [45], which has higher accuracy that ours on VOC tasks), the accuracy of our DLDL methods can be further improved.



(a) ChaLearn (b) Morph (c) Pointing'04 (d) AFLW pitch (e) AFLW yaw

Figure 6. Visualizations of hand-crafted and DLDL features using the t-SNE algorithm on *Morph*, *ChaLearn* and *AFLW* validation sets. The first row shows the embeddings of hand-crafted features (BIF or HOG). The second row shows the embeddings of the DLDL features derived from the penultimate fully connected layer of DLDL (best viewed in color).

D. Semantic segmentation

Datasets. We employ the PASCAL *VOC2011* segmentation dataset and the Semantic Boundaries Dataset (*SBD*) for training the proposed DLDL. There are 2,224 images (1,112 for training and 1,112 for testing) with pixel labels for 20 semantic categories in *VOC2011*. *SBD* contains 11,355 annotated images (8,984 for training and 2,371 for testing) from Hariharan *et al.* [49]. Following FCN [3], we train DLDL using the union set (8,825 images) of *SBD* and *VOC2011* training images. We evaluate the proposed approach on *VOC2011* (1,112) and *VOC2012* (1,456) test images.

Evaluation criteria. The performance is measured in terms of mean IU (intersection over union), which is the most widely used metric in semantic segmentation.

We keep the same settings as FCN including training images and model structure. The main change is that we employ KL divergence as the loss function based on label distribution (Eq. 15). Note that although we transform the ground-truth to label distribution in the training process, our evaluation rely only on ground-truth label.

Recently, Conditional Random Field (CRF) has been broadly used in many state-of-the-art semantic segmentation systems. We optionally employ a fully connected CRF [50] to refine the predicted category score maps using the default parameters of [51].

Results. Table VIII gives the performance of DLDL-8s and DLDL-8s-CRF on the test images of *VOC2011* and *VOC2012* and compares it to the well-known FCN-8s. DLDL-8s improves the mean IU of FCN-8s form 62.7% to 64.9% on *VOC2011*. On *VOC2012*, DLDL-8s leads to an improvement of 2.3 points in mean IU. DLDL achieves better results than FCN, which suggests it is important to improve the segmentation performance using label ambiguity. In addition, the CRF further improve performance of DLDL-8s, offering a 2.6% absolute increase in mean IU both on *VOC2011* and *VOC2012*.

Fig. 7 shows four semantic segmentation examples from the *VOC2011* validation images using FCN-8s, DLDL-8s and DLDL-8s-CRF. We can see that DLDL-8s can successfully segment some small objects (*e.g.*, car and bicycle) and particularly improve the segmentation of object boundaries (*e.g.*, horse's leg and plant's leaves), but FCN-8s does not. DLDL-8s may fail, *e.g.*, it sees a flowerpot as a potted plant in the fourth row in Fig. 7. Furthermore, compared to DLDL-8s, DLDL-8s-CRF is able to refine coarse pixel-level label predictions to produce sharp boundaries and fine-grained segmentations (*e.g.*, plant's leaves).

Table VIII
COMPARISONS OF DLDL AND FCN ON THE PASCAL *VOC2011* AND *VOC2012* TEST SETS.

Methods	mean IU VOC2011 test	mean IU VOC2012 test
FCN-8s [3]	62.7	62.2
DLDL-8s	64.9	64.5
DLDL-8s+CRF	**67.6**	**67.1**

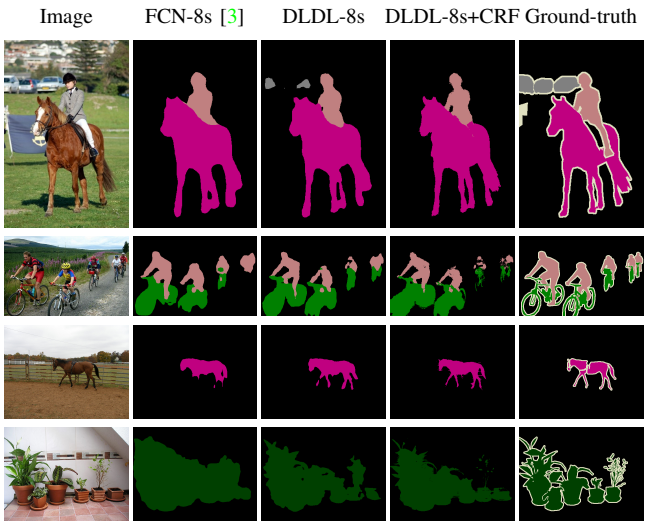
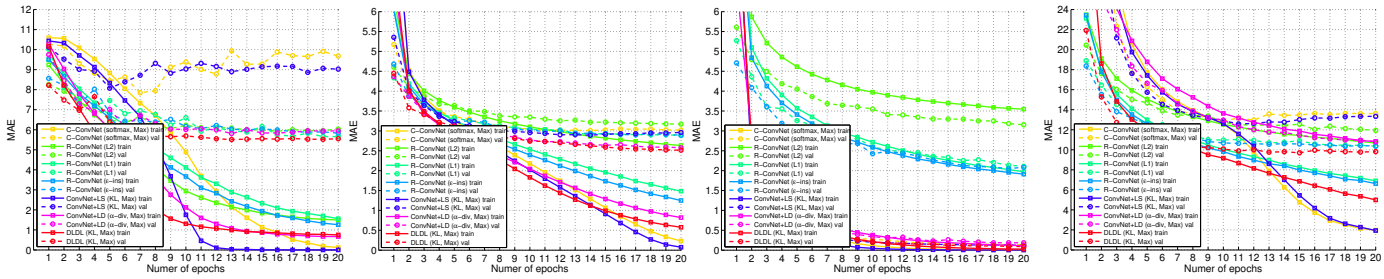


Figure 7. Semantic segmentation examples using FCN-8s, DLDL-8s and DLDL-8s-CRF on PASCAL *VOC2011* validation set.



(a) ChaLearn (b) Morph (c) BJUT-3D (d) AFLW

Figure 8. Comparisons of training and validation MAE of DLDL and all baseline methods on the *ChaLearn*, *Morph*, *BJUT-3D* and *AFLW* datasets (lower is better, best viewed in color).

V. Discussions

In this section, we try to understand the generalization performance of DLDL through feature visualization, and to analyze why DLDL can achieve high accuracy with limited training data. In addition, a study of the hyper-parameter is also provided.

Feature visualization. We visualize the model features in a low-dimensional space. Early layers learn low-level features (*e.g.*, edge and corner) and latter layers learn high level features (*e.g.*, shapes and objects) in a deep ConvNet [19]. Hence, we extract the penultimate layer features (4,096-dimensional) on *Morph*, *ChaLearn*, *Pointing'04* and *AFLW* validation sets. To obtain the 2-dimensional embeddings of the extracted high dimensional features, we employ a popular dimension reduction algorithm t-SNE [52]. The low-dimensional embeddings of validation images from the above four datasets are shown in Fig. 6. The first row shows the 2-dim embeddings of hand-crafted features (BIF for *Morph* and *Chalearn*, HOG for *Pointing'04* and *AFLW*) and the second row shows that of the DLDL features. These figures are colored by their semantic category. It can be observed that clear semantic clusterings (old or young for age datasets, left or right, up or down for head pose datasets) appear in deep features but do not in hand-crafted features.

Reduce over-fitting. DLDL can effectively reduce over-fitting when the training set is small. This effect can be explained by the label ambiguity. Considering an input sample X with one single label l. In traditional deep ConvNet, $y_l = 1$ and $y_k = 0$ for all $k \neq l$. In DLDL, the label distribution $\boldsymbol{y}$ contains many non zeros elements. The diversity of labels helps reduce over-fitting. Moreover, the objective function (Eq. 3) of DLDL can be rewritten as

$$T = -(y_l \ln \hat{y}_l + \sum_{k \neq l} y_k \ln \hat{y}_k)\,. \tag{20}$$

In Eq. 20, the first term is the tradition ConvNet loss function. The second term maximize the log-likelihood of the ambiguous labels. Unlike existing data augmentation techniques such as random cropping on the images, DLDL augments data on the label side.

In Fig. 8, MAE is shown as a function of the number of epochs on two age datasets (*ChaLearn* and *Morph*) and two head pose datasets (*BJUT-3D* and *AFLW*). On *ChaLearn* and *AFLW*, C-ConveNet (softmax) achieves the lowest training MAE, but produces the highest validation MAE. In particular, the validation MAE increases after the 8th epoch on *ChaLearn*. Similar phenomenon is observed on *AFLW*. This fact shows that over-fitting happens in C-ConvNet when the number of training images is small. Although there are 15,561 training images in *AFLW*, each category contains on averagely 4 training images since there are 3,721 categories.

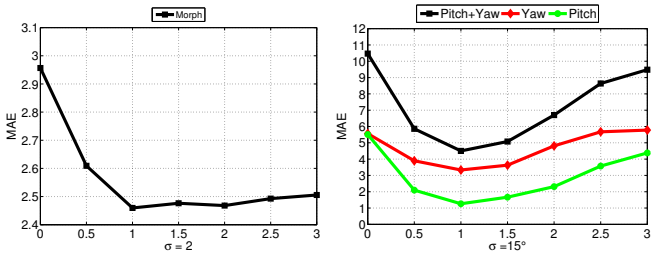


Figure 9. The performance (MAE) of DLDL with different label distributions (different parameter σ). The left figure is for the *Morph* dataset, while the right figure is for the *Pointing'04* dataset (lower is better).

Accelerate convergence. We further analyze the convergence performance of DLDL, C-ConvNet and R-ConvNet. We can observe that the training MAE is reduced very slowly at the beginning of training using C-ConvNet and R-ConveNet in many cases as shown in Fig. 8. On the contrary, the MAE of DLDL reduces quickly.

Robust performance. One notable observation is that C-ConvNet and R-ConveNet is unstable. Fig. 8c shows the MAE for pitch+yaw, a complicated estimation of the joint distribution. This is a very sparse label set because the interval of adjacent class (pitch or yaw) is $10°$. R-ConvNet has difficulty in estimating this output, yielding errors that are roughly 20 times higher than DLDL and C-ConvNet. On the other hand, C-ConvNet easily fall into over-fitting when there are not enough training data (*e.g*, Fig. 8a and Fig. 8d). The proposed DLDL is more amenable to small datasets or sparse labels than C-ConvNet and R-ConvNet.

Analyze the hyper-parameter. DLDL's performance may be affected by the label distribution. Here, we take age estimation (*Morph*) and head pose estimation (*Pointing'04*) for examples. σ is a common hyper-parameter in these tasks if it is not provided in the ground-truth. We have empirically

set $\sigma = 2$ in *Morph*, and $\sigma = 15°$ in *Pointing'04* in our experiments. In order to study the impact of σ, we test DLDL with different σ values, changing from 0 to 3σ with 0.5σ interval. Fig. 9 shows the MAE performance on *Morph* and *Pointing'04* with different σ. We can see that a proper σ is important for low MAE. But generally speaking, a σ value that is close to the interval between neighboring labels is a good choice. Because the shape of all curves are V-shape like, it is also very convenient to find an optimal σ value using the cross-validation strategy.

VI. Conclusion

We observe that current deep ConvNets cannot successfully learn good models when there are not enough training data and/or the labels are ambiguous. We propose DLDL, a deep label distribution learning framework to solve this issue by exploiting label ambiguity. In DLDL, each image is labeled by a label distribution, which can utilize label ambiguity in both feature learning and classifier learning. DLDL consistently improves the network training process in our experiments, by preventing it from over-fitting when the training set is small. We empirically showed that DLDL produces robust and competitive performances than traditional classification or regression deep models on several popular visual recognition tasks.

However, constructing a reasonable label distribution is still challenging due to the diversity of label space for different recognition tasks. It is an interesting direction to extend DLDL to more recognition problems by constructing different label distributions.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[4] X. Geng, C. Yin, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2401–2412, 2013.

[5] S. G. Kong and R. O. Mbouna, "Head pose estimation from a 2D face image using 3D face morphing with depth parameters," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1801–1808, 2015.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[7] X. Geng and Y. Xia, "Head pose estimation based on multivariate label distribution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1837–1842.

[8] C. Xing, X. Geng, and H. Xue, "Logistic boosting regression for label distribution learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4489–4497.

[9] Z. He, X. Li, Z. Zhang, F. Wu, X. Geng, Y. Zhang, M.-H. Yang, and Y. Zhuang, "Data-dependent label distribution learning for age estimation," *IEEE Transactions on Image Processing*, 2017, to be published, doi: 10.1109/TIP.2017.2655445.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[11] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2830–2838.

[12] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 617–624.

[13] J. Lu, V. E. Liong, and J. Zhou, "Cost-sensitive local binary feature learning for facial age estimation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5356–5368, 2015.

[14] B. Ahn, J. Park, and I. S. Kweon, "Real-time head orientation from a monocular camera using deep neural network," in *Asian Conference on Computer Vision*, 2015, pp. 82–96.

[15] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of International Conference on Learning Representations*, 2015, pp. 1–14.

[17] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference*, 2015, p. 6.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2014, pp. 818–833.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[21] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 689–692.

[22] K. Ricanek Jr and T. Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," in *International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 341–345.

[23] S. Escalera, J. Fabian, P. Pardo, X. Baró, J. Gonzalez, H. J. Escalante, D. Misevic, U. Steiner, and I. Guyon, "Chalearn looking at people 2015: Apparent age and cultural event recognition datasets and results," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–9.

[24] T. Minka, "Divergence measures and message passing," Microsoft Research, Tech. Rep. MSR-TR-2005-173, 2005.

[25] X. Geng, "Label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.

[26] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *European Conference on Computer Vision*, 2014, pp. 720–735.

[27] K.-Y. Chang and C.-S. Chen, "A learning framework for age rank estimation based on face images with scattering transform," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 785–798, 2015.

[28] D. Yi, Z. Lei, and S. Z. Li, "Age estimation by multi-scale convolutional network," in *Asian Conference on Computer Vision*, 2015, pp. 144–158.

[29] I. Huerta, C. Fernández, C. Segura, J. Hernando, and A. Prati, "A deep analysis on age estimation," *Pattern Recognition Letters*, vol. 68, pp. 239–249, 2015.

[30] R. Rothe, R. Timofte, and L. Gool, "DEX: Deep EXpectation of apparent age from a single image," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 252–257.

[31] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision*, pp. 1–14, 2016, doi:10.1007/s11263-016-0940-36.

[32] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *FG Net Workshop on Visual Observation of Deictic Gestures*, 2004, pp. 1–9.

[33] B. Yin, Y. Sun, C. Wang, and Y. Ge, "BJUT-3D large scale 3D face database and information processing," *Journal of Computer Research and Development*, vol. 46, no. 6, pp. 1009–1018, 2009.

[34] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2011, pp. 2144–2151.

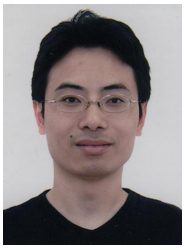
[35] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, no. 1–2, pp. 123–141, 1995.

[36] K. Sundararajan and D. Woodard, "Head pose estimation in the wild using approximate view manifolds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 50–58.
[37] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "CNN: single-label to multi-label," *CoRR, abs:1406.5726*, 2014.
[38] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3286–3293.
[39] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*, 2014, pp. 391–405.
[40] H. Yang, J. T. Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai, "Exploit bounding box annotations for multi-label object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 280–288.
[41] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," *CoRR, abs:1312.4894*, 2013.
[42] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1901–1907, 2015.
[43] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
[44] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan, "Instance-Aware hashing for multi-label image retrieval," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2469–2479, 2016.
[45] R.-W. Zhao, J. Li, Y. Chen, J.-M. Liu, Y.-G. Jiang, and X. Xue, "Regional gating neural networks for multi-label image classification," in *Proceedings of the British Machine Vision Conference*, vol. 6, 2016.
[46] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, and S. Yan, "Subcategory-aware object classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 827–834.
[47] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan, "Contextualizing object detection and classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1585–1592.
[48] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1717–1724.
[49] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 991–998.
[50] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with gaussian edge potentials," in *Advances in Neural Information Processing Systems*, 2011, pp. 109–117.
[51] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proceedings of International Conference on Learning Representations*, 2015.
[52] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

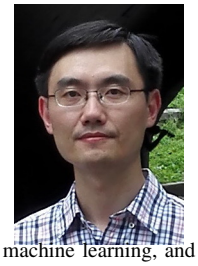
Bin-Bin Gao received the B.S. and M.S. degrees in applied mathematics in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Technology, Nanjing University, China. His research interests include computer vision and machine learning.

Chao Xing received the B.S. degree in software engineering from Southeast University, China, in 2014. He is currently a postgraduate student in the School of Computer Science and Engineering at Southeast University, China. His research interests include pattern recognition, machine learning, and data mining.

Chen-Wei Xie received his B.S. degree from Southeast University, China, in 2015. He is currently a postgraduate student in the Department of Computer Science and Technology, Nanjing University, China. His research interests include computer vision and machine learning.



Jianxin Wu (M'09) received the B.S. and M.S. degrees in computer science from Nanjing University, and the Ph.D. degree in computer science from the Georgia Institute of Technology. He was an Assistant Professor with the Nanyang Technological University, Singapore. He is currently a Professor with the Department of Computer Science and Technology, Nanjing University, China, and is associated with the National Key Laboratory for Novel Software Technology, China. His current research interests include computer vision and machine learning. He has served as an Area Chair for CVPR 2017 and ICCV 2015, a Senior PC Member for AAAI 2017 and AAAI 2016, and an Associate Editor of *Pattern Recognition Journal*.



Xin Geng (M'13) received the B.S. and M.S. degrees in computer science from Nanjing University, China, in 2001 and 2004, respectively, and the Ph.D degree from Deakin University, Australia in 2008. He joined the School of Computer Science and Engineering at Southeast University, China, in 2008, and is currently a professor and vice dean of the school. He has authored over 50 refereed papers, and he holds five patents in these areas. His research interests include pattern recognition, machine learning, and computer vision.