Adaptively Weighted Multi-task Deep Network for Person Attribute Classification

Keke He[1,2], Zhanxiong Wang[1,2], Yanwei Fu[1,3*]
Rui Feng[1,2], Yu-Gang Jiang[1,2], Xiangyang Xue[1,2,3]
[1]Shanghai Key Laboratory of Intelligent Information Processing;
[2]School of Computer Science, Fudan University; [3]School of Data Science, Fudan University
{kkhe15,15210240046,yanweifu,fengrui,ygj,xyxue}@fudan.edu.cn

ABSTRACT

Multi-task learning aims to boost the performance of multiple prediction tasks by appropriately sharing relevant information among them. However, it always suffers from the negative transfer problem. And due to the diverse learning difficulties and convergence rates of different tasks, jointly optimizing multiple tasks is very challenging. To solve these problems, we present a weighted multi-task deep convolutional neural network for person attribute analysis. A novel validation loss trend algorithm is, for the first time proposed to dynamically and adaptively update the weight for learning each task in the training process. Extensive experiments on CelebA, Market-1501 attribute and Duke attribute datasets clearly show that state-of-the-art performance is obtained; and this validates the effectiveness of our proposed framework.

KEYWORDS

facial attribute analysis, person attribute analysis, deep learning, multi-task learning

ACM Reference Format:
Keke He[1,2], Zhanxiong Wang[1,2], Yanwei Fu[1,3] and Rui Feng[1,2], Yu-Gang Jiang[1,2], Xiangyang Xue[1,2,3] . 2017. Adaptively Weighted Multi-task Deep Network for Person Attribute Classification . In *Proceedings of MM '17, Mountain View, CA, USA, October 23–27, 2017,* 9 pages.
https://doi.org/10.1145/3123266.3123424

1 INTRODUCTION

The techniques of recognizing person attributes have attracted great research attention over the past several decades. As an umbrella term, person attributes refer to the attributes of people, such as facial and clothing attributes [21], and biological traits [9] (*e.g.* age and gender). These person attributes can either serve as the middle-level features for high-level computer vision tasks such as person re-identification, or be directly used for advanced multimedia applications, *e.g.*, clothing recommendation. The techniques of analyzing these person attributes also facilitate a wide range of real-world applications such as financial monitoring, building access control, station ticket check system and so forth. For example, the techniques can automatically identify or verify individual's identity by examining his/her facial and personal traits/attributes from the physiological and/or visual appearance characteristics, rather than authenticating persons by the other assistant methods such as passwords, PINs, tokens and so on. To efficiently boost the performance and improve the generalization ability of task learners, the multimedia system of recognizing person attributes jointly train the learner on predicting each person attribute. This is inspired by the ability of *learning to learn* [3, 28] of humans. Particularly, the multi-task learning can seamlessly and effectively learn, transfer and share the knowledge across multiple related tasks. However, it is also an open challenge of learning to leverage the relationships among different learning tasks.

Figure 1: The examples of attribute classification. The left image pair is "hat" attribute, the right image pair is "smiling" attribute. Obviously, it will be easier to classify "hat" attribute than "smiling", thus result in different tasks have various learning difficulties.

With the renaissance of convolutional neural networks (CNN), deep multi-task network has been developed to learn to share the representations across different categories. The natural strategy is to build a multi-task network that enables sharing the model parameters in all layers except the last layer which learns to predict each individual task. Though this is simple enough, such a sharing strategy, unfortunately may suffer from the problem of *negative transfer*: when two tasks are dissimilar, the inadequate brute-force transfer may hurt the learner's performance. One common practice towards alleviating this problem is to increase the number of top layers which can exclusively model each individual task [8, 11, 16, 23]. The intuition behinds this is that the network can enforce the bottom layers to share low-level information [32], and learn task-specific sub-networks on top layers. However, the searching space of configuring multi-task deep architectures is combinatorially large, and quite often the designed network is biased

*Yanwei Fu is the corresponding authour. Email: yanweifu@fudan.edu.cn

MM '17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123424

by the designer's perception of the relationship among different tasks as pointed out in [21].

In addition to this problem, the state-of-the-art multi-task learning works keep the weights fixed for each learning task, rather than *dynamically* and *adaptively* changing them. Different tasks of learning person attributes inherently have different learning difficulties, as well as varying convergence rates. For example, learning to identify the "hat" attribute is much easier than estimating whether one face is "smiling" in Fig 1.

To solve these two problems, this study aims at investigating an inbuilt sharing mechanism that is possible of *dynamically* and *adaptively* coordinating the relationships of learning different person attribute tasks. Critically, the weighted loss layer is defined to model the relationship of learning different person attribute tasks. The weight of each task should be *dynamically* tuned in the training stages. To formally implement this idea, we propose an *adaptively* weighted multi-task deep framework to jointly learn multiple person attributes, and a validation loss trend algorithm to automatically update the weights of weighted loss layer. The weights are dynamically changed in the training process according to the generalization ability of each task learner, which are approximately measured by the validation set. A validation loss trend algorithm is proposed to adaptively update the weights of this layer. Extensive experiments on benchmark including CelebA [20], Market-1501 attribute [17, 40] and Duke-attribute datasets [17, 41] demonstrate that our proposed framework significantly outperforms the state-of-the-art alternatives.

Contributions. To the best of our knowledge, we for the first time, propose a novel sharing mechanism of jointly learning multi-tasks of person attributes. The mechanism is capable of *dynamically* and *adaptively* weighting the "importance" of each task by the corresponding learner's generalization ability. To efficiently encode and implement this mechanism, our deep multi-task network utilizes the weighted loss layer; and we propose a simple yet effective approach – validation loss trend algorithm, which can validate the loss trend in the training process and automatically and adaptively tune the weights of learning each attribute task. Experimentally, we illustrate the efficacy of proposed framework on multi-task learning on person attribute analysis.

2 RELATED WORK

2.1 Multi-task Learning

The proposed framework belongs to the category of Multi-task Learning (MTL) which is one type of transfer learning [22, 32]. The MTL exploits the shared information among several different tasks on the same dataset. The recent deep models are especially good for MTL since the extracted hierarchical features of one task may be very useful for other tasks [25, 42]. Thus MTL can facilitate solving many tasks and applications such as pose estimation and action recognition [43], semantic classification and information retrieval [19], facial landmark detection [23, 36], and the prediction of person attributes [1, 6, 21, 24, 26].

The deep multi-task models have been investigated in several recent works, e.g., the Hyperface [23, 36, 37]. To solve the facial landmark detection problem, Zhang *et al.* [36, 37] jointly optimized the facial landmark detection with other heterogeneous but subtly correlated tasks such as gender, expression, and appearance attributes. Their goal is thus to employ the auxiliary tasks of predicting attributes to help gain the performance for the main task – facial landmark detection. Our work is different from [36, 37] in that we aim at learning to predict person attributes, rather than using auxiliary attributes to help learn the task of facial landmark detection. Thus our validation trend loss proposed does not necessarily split the main and auxiliary tasks.

Zhang *et al.* [34] proposed a deep cascaded multi-task framework which exploited the inherent correlation between face detection and alignment to boost up their performance. Jou *et al.* [12] proposed a multi-task cross-residual network for knowledge transfer. Abdulnabi *et al.* [1] proposed a multi-task CNN model to allow sharing of visual knowledge between tasks to learn facial attributes. Unfortunately, these networks have to rely on hand-designed architectures rather than dynamically and adaptively weighting each task learner as our work in the training process.

Lu *et al.* [21] addressed the problem of dynamically designing a compact and adaptively changed multi-task deep architectures by grouping similar tasks. In contrast, our work focuses on dynamically and adaptively optimizing the weight of each learner.

2.2 Person Attribute Analysis

The works of recognizing attributes of persons such as facial and clothing attributes have received great attention during the past few years. In visual surveillance scenario, the facial and person attribute prediction can enable many practical applications such as searching based on semantically descriptions [14, 27], and interpreting the face verification or person re-identification results in a human-comprehensible form [15]. In general, the person attributes can be utilized as mid-level features for improving other tasks, for example, person re-identification [4, 17, 39], clothing retrieval [5, 29], and fashion recommendation [18].

The tasks of facial attribute prediction has achieved a series of breakthroughs recently by employing the deep models [20, 30, 42]. Facial attributes have also been shown to be helpful in improving face detection [38], face alignment [31] and face identification [14]. Rudd *et al.* [26] introduced a mixed objective optimization network which took account of different distributions of attribute labels. Facial attributes have been directly used to build the face identification classifier in [14].

3 METHODOLOGY

The adaptively weighted multi-task deep convolutional neural network is proposed in this section. The framework is overviewed in Fig. 2. It is composed of four types of layers, namely, *convolutional layers, fully connected layers, building block layers* and *weighted loss layer*. The basic network structure includes the convolutional, fully connected and building block layers. This basic network is shared and used for all attribute prediction tasks. In this work, ResNet-50 [7] is employed as the basic network. The weighted loss layer is newly proposed here and will be described in Sec. 3.3.

We propose a novel training algorithm – a validation loss trend algorithm in Sec. 3.4, to jointly learn the multiple attribute prediction

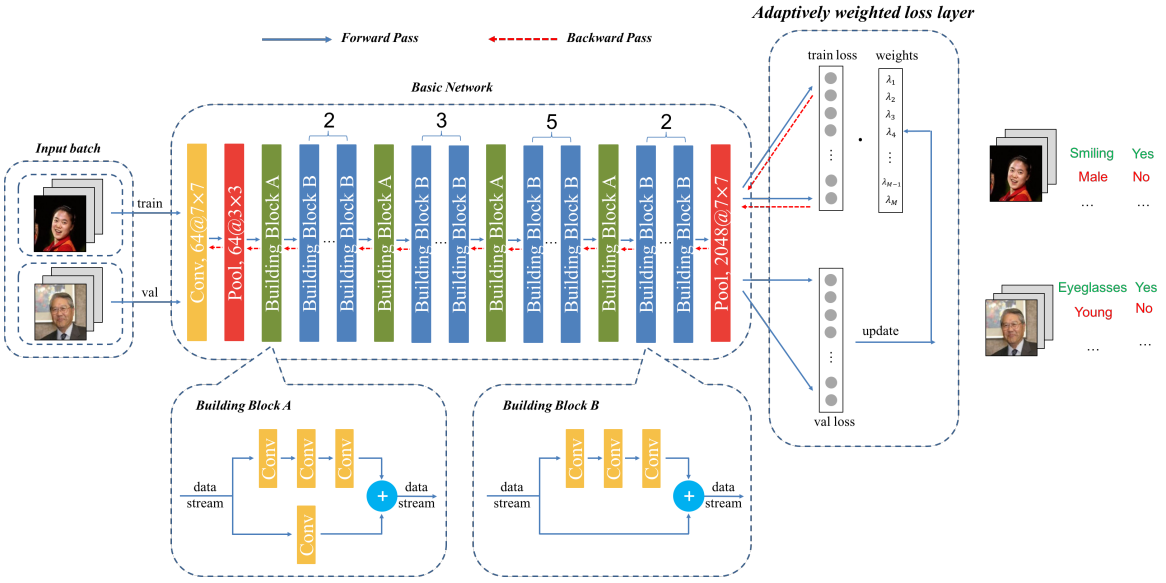


Figure 2: Overall of our network. The basic network is ResNet-50 [7]. Note that: (1) the 2, 3, 5 and 2 indicates the corresponding number of building blocks. (2) number@size indicates the number and the size of filters in convolutional layers.

tasks simultaneously. The validation loss trend algorithm is conducted in the backward pass of the Back-Propagation (BP) algorithm in training the deep network in an end-to-end way. Specifically, the forward pass propagates the input vectors forward through the network, layer by layer, until it reaches the output layers of each attribute prediction task. Each output value is compared against the desired output by a loss function as the training error. Rather than treating each prediction task equally, our validation loss trend algorithm can dynamically update the weight of each task learner. In the backward pass, the weighted loss (described in Sec. 3.3) is propagated to update the parameters of the basic network.

3.1 Problem Setup

Our goal is to learn the predictors that can confidently and effectively predict the existence of attributes of person images. Suppose we have the labelled source training dataset $\mathcal{D}_s = \{\mathbf{I}, \mathbf{a}, \mathbf{L}\}$ with N training instances and M attributes. $\mathbf{I}$ denotes the training images, $\mathbf{a}$ is the attributes and $\mathbf{L}$ denotes the labels. If the i-th image $\mathbf{I}_i$, $(i = 1, \cdots, N)$ is annotated to have the j–th attribute $\mathbf{a}_j$ $(j = 1, \cdots, M)$, we denote $\mathbf{L}_{ij} = 1$; otherwise, $\mathbf{L}_{ij} = -1$. Given a new test image $\mathbf{I}^\star$, the goal is then to learn a function $\mathbf{a}^\star = \Psi\left(\mathbf{I}^\star\right)$ using all available training information and predict the attribute vector $\mathbf{a}^\star$. Note that since each image can be labelled with multiple attributes, we have the predicting functions $\Psi = \left[\psi_j\right]_{j=1,\cdots,M}$, and $\psi_j\left(\mathbf{I}^\star\right) \in \{+1, -1\}$.

3.2 Separate Model and Basic Model

We firstly introduce our naive baselines – separate model and basic model.

Separate model. To enable our task, we can train an independent binary classifier for each individual attribute as done in [20]. This can be modeled as minimizing the expected loss over all the training instances for the j-th attribute $\mathbf{a}_j$; and it leads to the following formulation as,

$$\Theta_j = \text{argmin}_{\Theta_j} \sum_{i=1}^{N} \mathcal{L}\left(\psi_j\left(\mathbf{I}_i; \Theta_j\right) - \mathbf{L}_{ij}\right) \tag{1}$$

where Θ_j in Eq (1) indicates the optimized parameter set of the j-th attribute prediction network; and $\mathcal{L}(\cdot)$ is the loss function penalized the value differences of predicted attributes and ground-truth attributes. The $\mathcal{L}(\cdot)$ can be square error loss to make a more fair comparison with [26]. Note that hinge-loss [15, 20] can also be used for $\mathcal{L}(\cdot)$. In our experiments, there is no significant difference of these two types of loss functions; and our experimental conclusions are still held.

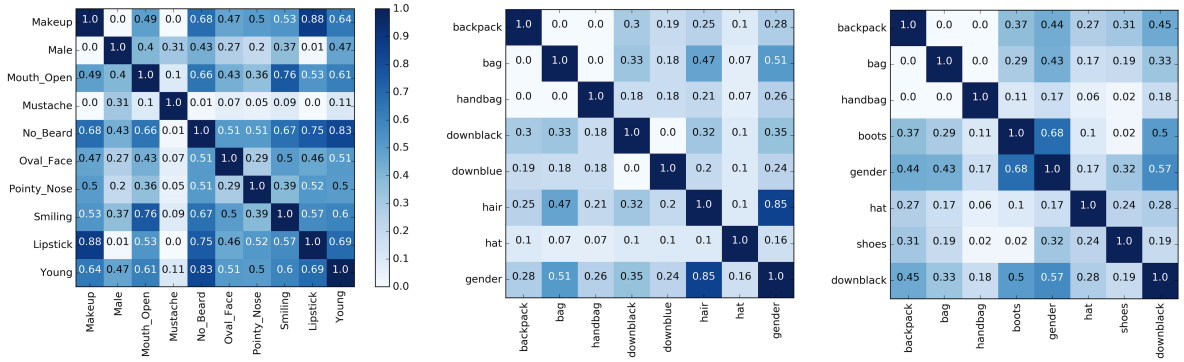
Basic model. To jointly optimize all the attribute prediction tasks, we extend the model of Eq (1) as,

$$\Theta = \text{argmin}_{\Theta} \sum_{j=1}^{M} \sum_{i=1}^{N} \mathcal{L}\left(\psi_j(\mathbf{I}_i; \Theta) - \mathbf{L}_{ij}\right) \tag{2}$$

where the same parameter set Θ are shared across all the prediction tasks.

3.3 Our Model with Weighted Loss Layer

Eq (1) considers each attribute independent; and in Eq (2) each attribute prediction task contributes equally to the expected loss. In contrast, we believe that each attribute prediction task should neither be a standalone problem as the separate model, nor equally weighted as the basic model. Intuitively, one person attribute may be well correlated with another one. For example, it is highly likely that the attribute "lipstick" should be positively correlated with the "bushy eyebrows", and negatively related to the attribute "mustache",



(a) CelebA dataset (b) Market-1501 attribute dataset (c) Duke attribute dataset

Figure 3: Correlation map of random selected attributes on CelebA, Market-1501 attribute and Duke attribute dataset. We can observe that some attributes are highly positive or negative correlated with other attributes.

since a beautiful lady is more prone to use "lipstick" and has "bushy eyebrows" without "mustache".

To further illustrate this point, we randomly select 10 different facial attributes from CelebA dataset to compute their correlations by using the ground-truth attribute annotations. The visualized results in Fig. 3 clearly validates our argument. For example, the correlation between "lipstick" and "makeup" is 0.88, and "mouth_open" is positively correlated with "smiling" by 0.76. Furthermore, different tasks of learning person attributes inherently have different learning difficulties, as well as varying convergence rates. For example, learning to identify the "hat" attribute is much easier than estimating whether one face is "smiling". Thus, rather than enforcing our network to learn all tasks with equal force, we adaptively weight the tasks learned as follows,

$$\Theta = \text{argmin}_{\Theta} \sum_{j=1}^{M} \sum_{i=1}^{N} \left\langle \lambda_j, \mathcal{L}\left(\psi_j(\mathbf{I}_i; \Theta) - \mathbf{L}_{ij}\right) \right\rangle \tag{3}$$

where $\langle \cdot \rangle$ indicates the operation of the inner product; λ_j is the scalar value to weight the importance of the task of learning j–th attributes. The values of λ_j can be used to construct the weighted loss layer as illustrated in Fig. 2.

In the testing stage, we can binarize the prediction results of the testing image $\mathbf{I}_k$ for the j–th attribute $\mathbf{a}_j$ $(j = 1, \cdots, M)$ and predict the attribute annotation $\hat{\mathbf{L}}_{kj}$ as,

$$\hat{\mathbf{L}}_{kj} = \begin{cases} 1 & \psi_j(\mathbf{I}_k) > \tau \\ -1 & \psi_j(\mathbf{I}_k) \leqslant \tau \end{cases} \tag{4}$$

where τ is the threshold parameter.

3.4 Validation Loss Trend Algorithm

3.4.1 Problem of tuning λ_j in Eq (3). From the formulations of Eq (3), it is quite clear that the hyper-parameter $\lambda_j (j = 1, \cdots, M)$ is the key of our network – once the weight of each task is set, the parameter set Θ can be updated accordingly and the estimation of attribute annotation $\hat{\mathbf{L}}_{kj}$ becomes straightforward. However, tuning the weight parameter $\lambda_j (j = 1, \cdots, M)$ in Eq (3) is very difficult. Traditional methods such as cross validation is not applicable here due to the huge search space and prohibitive cost of training networks. For example, the 40 attributes on CelebA dataset means that if using cross validation, we have to compute 40 different λ_j $(j = 1, \cdots, 40)$.

Algorithm 1 Network Training Procedure. c is the current training iteration, λ is the weights of all the tasks, *val_loss_list* is a data structure to save the validation loss.

Require: k: the weight updating period.
Initialize: $c = 0$, $\lambda = \mathbf{1}$, $val\_loss\_list = []$.
while $c < max\_iter$ **do**
  $train\_loss, val\_loss \leftarrow net.feedward()$
  $val\_loss\_list.append(val\_loss)$
  if $c\%k = 0$ **then**
    $\lambda = update\_weights()$
  end if
  $weighted\_loss \leftarrow train\_loss * \lambda$
  $net.backward(weighted\_loss)$
  $c = c + 1$
end while

Algorithm 2 Update weights of tasks: *update_weights*()

Require: k is the weight updating period, c is the current training iteration, *val_loss_list* is a data structure to save validation loss.
Initialize: $\lambda = \mathbf{1}$
if $c >= 2 * k$ **then**
  $pre\_mean \leftarrow mean(val\_loss\_list[c - 2 * k : c - k])$
  $cur\_mean \leftarrow mean(val\_loss\_list[c - k : c])$
  $trend \leftarrow abs(cur\_mean - pre\_man)/cur\_mean$
  $norm\_trend \leftarrow trend/mean(trend)$
  $norm\_loss \leftarrow cur\_mean/mean(cur\_mean)$
  $\lambda \leftarrow norm\_trend * norm\_loss$
  $\lambda \leftarrow \lambda/mean(\lambda)$
end if
return λ

3.4.2 Validation loss trend algorithm. To automatically compute the parameters λ_j, we propose an efficient and yet effective

approach to validate the loss trend in the training process in order to automatically and adaptively coordinate the importance of learning each attribute task. The intuition behind our validation loss trend algorithm is that in learning multiple tasks simultaneously, the "important" tasks should be given high weight (*i.e.* λ_i) to increase the scale of loss of the corresponding tasks.

Nonetheless, it is also nontrivial to directly measure the importance of one task. One common practice is to manually define the main and auxiliary tasks as has been done in [36, 37]; and the main task is the important one and should always be optimized in a high priority; those auxiliary tasks are in low priority and should be stopped if these tasks hurt the performance of the main task learner. In general, on person attribute analysis, we can not have the pre-specified important degree of each task.

As an alternative, the generalization ability is served as an objective measurement of the "importance" of one task. Specifically, when multiple tasks have been learned, the trained model of one task with lower generation ability should be set higher weight than those models of the other tasks. The generalization ability of one learned model can be measured by the validation set, which is kept unknown in the training process.

We thus formulate the validation loss trend algorithm to jointly learn multiple tasks in Alg. 1. The main advantage of our algorithm is to adaptively learn the weights of all the tasks by the weighted loss layer. Suppose c be the current iteration of updating the network parameters Θ and λ is initialized as an all ones vector.

As illustrated in Fig. 2, in each batch, we sample the images from training and validation set respectively. For example, each batch has 10 training images and 10 validation images[1]. *Only the training images can be used to update the network parameters* Θ.

The validation loss is computed in each training iteration by the 10 validation images; the weight vector λ is updated every k iterations by the $update\_weights()$ (in Alg. 2). The updated weights are then utilized to compute the loss of training data and update the network parameters Θ in the backward pass.

The way of updating λ (i.e. $update\_weights()$) is explained in Alg 2. Since 10 validation images are used to compute the validation loss, we compute the "general" trend of validation loss over k iterations. We compute the mean validation loss with both current and previous k iterations, as cur_mean and pre_mean (in Alg 2) respectively. The trend of validation loss of tasks is computed as $trend \leftarrow abs(cur\_mean - pre\_man)/cur\_mean$. The $trend$ is a M dimension vector. The weight vector λ thus should be decided by these two key factors – normalized trend ($norm\_trend$) and normalized validation loss ($norm\_loss$).

4 EXPERIMENTS

4.1 Datasets and Settings

We present an extensive evaluation of our approach on multi-task person attribute prediction. We use CelebA [20] dataset for facial attribute tasks; and Market1501-attribute [17, 40] and Duke-attribute dataset [17, 41] for pedestrian attribute tasks.

CelebA is a facial attribute dataset of approximately $200k$ images of $10k$ identities [20]. Each image is annotated with 5 landmarks(two eyes, the nose tips, the mouth corners) and binary labels of 40 attributes. In this paper, to make a fair comparison with the other facial attribute methods, we use the standard split: the first $160k$ images are used for training, $20k$ images for validation and remaining $20k$ for testing. CelebA provides the pre-cropped face images by 5 landmarks, we use cropped training images to train attribute prediction models and test models same as other methods [26].

Market-1501 attribute dataset [17] is an extension of Market-1501 dataset [40] with person attribute annotations. The dataset contains 32,688 images of 1501 identities. Among them 751 and 750 identities are used as the training and testing set respectively. It contains 12 different types of annotated attributes, including 10 binary attribute (such as gender, hair length, and sleeve length) and 2 multi-class attributes, *i.e.* colors of upper and lower body clothing.

Duke attribute dataset [17] contains 1812 identities captured under 8 cameras. Training and testing set both have 702 identities with 16522 training images and 17661 testing gallery images respectively. It is annotated with 8 binary pedestrian attributes such as wearing a hat, and wearing boots, and 2 multi-class attributes.

Evaluation metrics. The person attribution prediction can be taken as the problems of classifying many attributes. The standard classification accuracy of each attribute as well as the mean accuracy over all attributes are utilized to evaluate our performance. These metrics have also been used by previous work [17, 21].

Parameter settings. We use the open source deep learning framework Caffe [10] to implement the weighted loss layer and train our multi-task network. For all the experiments, we only use a single end-to-end model for all the testing. The τ of Eq (4) is set as 0. Square error loss is used in Eq (3) as well as the corresponding competitors in order to make a fair comparison with [26]. The weight changing interval k is set as 200 in Alg.1. We use the stochastic gradient descent algorithm. Dropout is used for fully connected layers and the ratio is set to 0.5. (1) On CelebA dataset, to train the facial attribute model, the base learning rate is set as 0.0001, and gradually decreased by 1/10 at $70k$, $100k$, $120k$. The input image is resized to $224 \times 224 \times 3$, the mini-batch size is set as 20. (2) On the Market-1501 and Duke attribute datasets, we use slightly less training iterations due to the relatively small number of training instances: the base learning rate is still set as 0.0001, and gradually decreased by 1/10 at $10k$, $15k$.

Running costs. Our face model get converged with $130k$ iterations and it takes 14 hours on CelebA with one NVIDIA TITAN X GPU. Our pedestrian model get converged with $20k$ iterations and it takes 2 hours on Market-1501 and Duke with one NVIDIA TITAN X GPU. For training all the model, the batch size is 20, and it takes around 11 GB GPU memory.

4.2 Competitors

We list the competitors on each dataset. Note that please refer to the supplementary material for the full comparison results on each attribute.

Competitors on CelebA dataset. We firstly compare our framework with two baseline models (i.e. Separate models and Basic model) to validate the effectiveness of our multi-task framework; secondly, our results are also compared against those of the current

[1] Of course, the portion of training and validation images as well as batch size could be varied in different learning tasks.

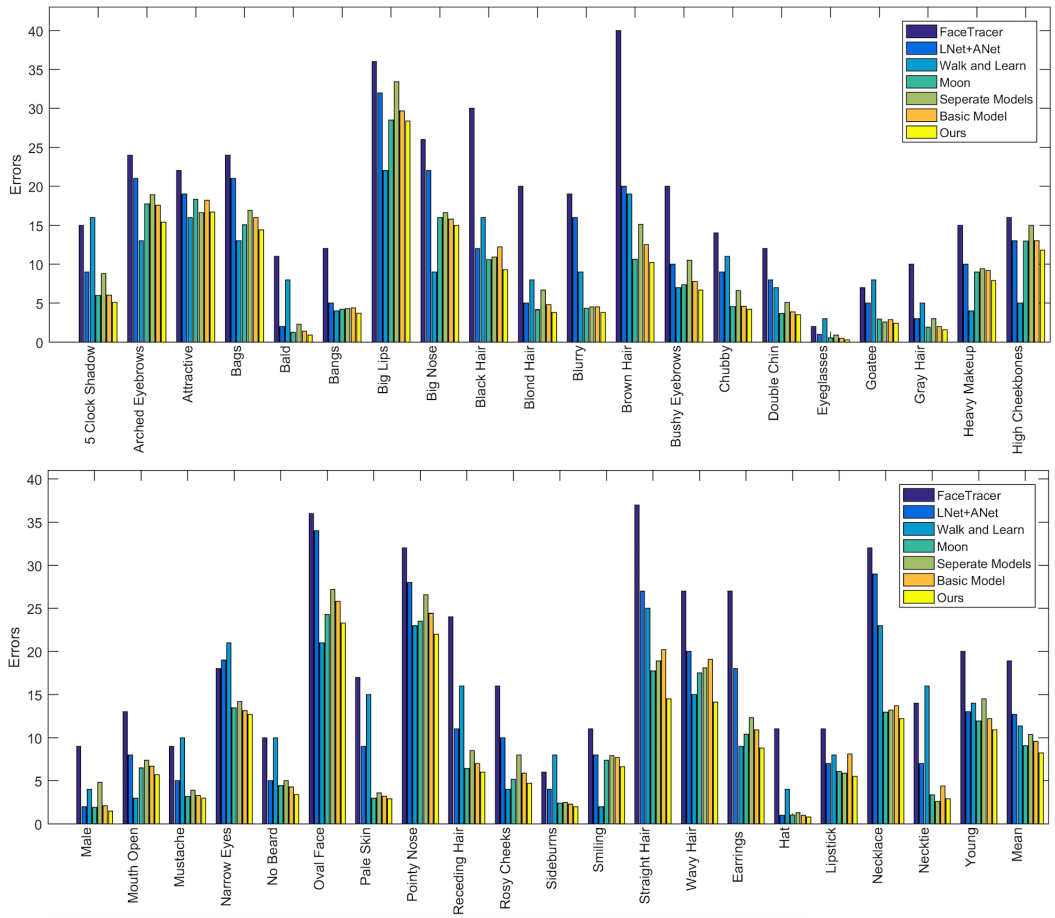


Figure 4: Performance comparison with state-of-the-art methods on CelebA dataset. *Error* = 1 − *Accuracy*.

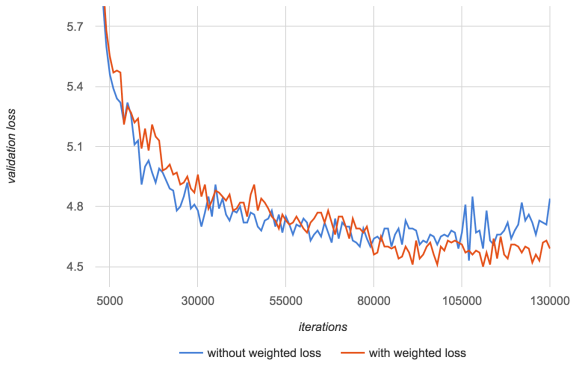


Figure 5: Comparison of with and without adaptively weighted loss in the training stages. The model using weighted loss clearly converges better.

Methods	Accuracy (%)
FaceTracer [13]	81
PANDA-w [35]	79
PANDA-l [35]	85
LNets+ANet [20]	87
MT-RBM-PCA [6]	87
Off-the-Shelf CNN [42]	86.6
Walk-and-Learn [30]	88
Moon [26]	90.94
Adaptive Sharing [21]	91.26
Separate Models	89.63
Basic Model	90.42
Our Model	**91.80**

Table 1: Comparison of mean accuracy on CelebA.

state-of-the-arts in order to evaluate the performance of our validation loss trend algorithm. Specifically, (1) *Separate Models* [33] trains each network model for each attribute. It results in totally 40 models trained. (2) *Basic model*, as explained in Eq(2), is used to train one single deep network model for all attributes; and this indicates all attributes use equal weight. (3) FaceTracer [13] is the best non-neural-network based algorithm so far; it extracts the HOG [2] and color histograms in several important functional face regions and

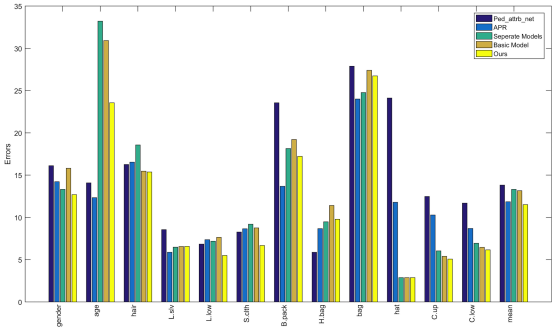


Figure 6: Performance comparison with state-of-the-art methods on Market-1501 attribute dataset. "L.slv", "L.low", "S.clth", "B.pack", "H.bag", "C.up", "C.low"denote length of sleeve, length of lower-body clothing, style of clothing, backpack, handbag, color of upper-body clothing and color of lower-body clothing, resp. $Error = 1 - Accuracy$**.**

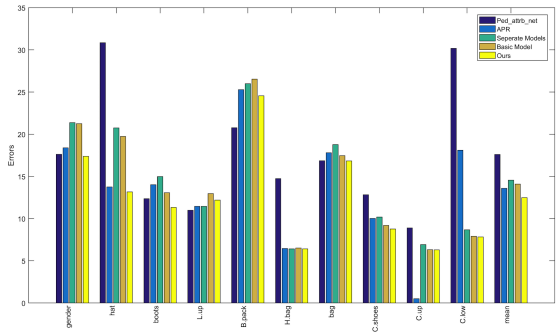


Figure 7: Performance comparison with state-of-the-art methods on Duke attribute dataset. $Error = 1 - Accuracy$**.**

Methods	Market-1501 (%)	Duke (%)
Ped_attrb_net [17]	86.19	82.39
APR [17]	88.16	86.42
Separate Models	86.68	85.45
Basic Model [35]	86.84	85.91
Our Model	**88.49**	**87.53**

Table 2: Comparison of mean recognition accuracy on Market-1501 and Duke attribute datasets.

then trains SVM for each attribute classification. (4) *PANDA-w* [35] and *PANDA-l* [35] are two variants of PANDA model [35]. As in [20], we also compare these two methods on facial attribute prediction. (5) *LNets+ANet* [20] combines two deep networks (including two face localization networks and one attribute extraction network), and uses SVM classifier to learn facial attributes. (6) *MT-RBM-PCA* [6] adopts a new multi-task restricted Boltzmann machine that models the distributions of multiple attributes and classifies them. (7) *Off-the-Shelf CNN* [42] uses off-the-shelf architectures trained for face recognition to build facial descriptors, then trains classifiers for attribute learning. (8) *Walk and Learn* [30] learns good representations for facial attributes by exploiting videos and contextual data as the person walks. (9) *Moon* [26] is a mixed objective optimization multi-task network to learn all facial attributes and advances face attribute recognition by learning multiple attribute labels simultaneously. (10) *Adaptive Sharing* [21] learns a deep MTL framework which can dynamically group similar tasks together. Note that only the mean accuracy results of (4), (6), (7) and (10) are reported here.

Competitors on two pedestrian attribute datasets. Several baselines are compared against our method on these two datasets to evaluate the effectiveness of our contributions. *(1) Separate Models,* we still train each CNN model for each attribute. (2) *Basic model* , we still train one CNN model for all the attributes with the weights of all attributes set to 1 as in Eq (2). (3) *APR* [17] is an attribute-person recognition network which learns a discriminative embedding for person re-ID and attribute predictions. (4) *Ped_attrb_net* is short for pedestrian attribute recognition network, which is a baseline method used in [17] to evaluate the performance of pedestrian attribute recognition.

4.3 Results on CelebA Dataset

We evaluate the facial attribute prediction tasks with the standard settings of CelebA dataset. The results are showed in Fig 4. We highlight the following observations.

(1) The results of our model beat all the state-of-the-art methods. Comparing with all the other baselines, we highlight that ours achieves the best performance with the mean accuracy 91.80% over 40 facial attributes. In particular, our model can beat most state-of-the-art results on 39 attributes from all 40 facial attributes with the only exception – *Walk and Learn*, which is pre-trained on external data and we can beat on 29 attributes. This validates the efficacy of our adaptively weighted multi-task network. We also list the mean accuracy of different methods in Tab. 1. Among all these methods, Adaptive Sharing and Moon are two second best methods with the mean average accuracy 91.26% and 90.94% respectively on CelebA dataset. In contrast, our model achieves 91.80% accuracy, outperforming the Adaptive Sharing and Moon methods by clear margins 0.54% and 0.86%. It is also worth noticing that such an improvement of our method over Adaptive Sharing and Moon is statistically significant to show the effectiveness of our framework, since the prediction accuracies on most attributes are very high. For instance, as can be seen in Fig. 4, the accuracies of 17 attributes of Moon and our method are higher than 95%.

(2) The efficacy of weighted multi-task network. The results are compared in Fig. 4 and Tab. 1. As we can see from the figure and table, our model can obtain the mean accuracy of 91.80%; and it shows 2.17% and 1.38% improvements over the other two multi-task baselines – separate models and basic model respectively. This is due to the fact that our adaptively weighted loss layer can efficiently model the correlations of different attributes; and more critically, our validation loss trend algorithm can dynamically and automatically enforce heavy weights on the learning attribute tasks of relatively low performance on training and validation sets.

(3) Our results are also better than those works in [23, 30]. Additionally, we also report mean accuracy results of CelebA from the recent works – Walk-and-Learn [30] in Tab. 1. We note that our

model can beat this method by clear margins (*i.e.* 3.8%) due to the effectiveness of validation loss trend algorithm. Furthermore, on gender attribute prediction, our framework can hit 98.5% accuracy; in contrast, the accuracy of HyperFace [23] is 97.0%. That indicates that our results can get 1.5% improvement over that of HyperFace [23], again thanks to our adaptively weighted layer and validation loss trend algorithm.

Qualitatively results. The last row of Fig. 8 gives some examples of the attributes on CelebA dataset. Each column is corresponding to one type of attribute. In Fig. 5, we visualize the changes of validation loss versus the number of iterations of our method and the basic model. As we can read from the figure, both methods have almost similar validation loss in the first 80, 000 iterations. As the number of iterations is beyond 80, 000, our validation loss is decreased even further, and yet the loss of basic model is kept steady. This reveals that our validation loss trend algorithm can better optimize the attribute prediction tasks of lowest generalization ability.

4.4 Results on Pedestrian Attribute Datasets

We evaluate the pedestrian attribute prediction tasks on Market-1501 attribute and Duke attribute datasets.

(1) The results of our model beat all the state-of-the-art methods on both two datasets. For Market-1501 dataset, our framework achieves the mean accuracy of 88.49% over 12 pedestrian attributes, outperforming the current best method (APR [17]) by 0.33% absolute percentage points. In particular, our model can beat all the state- of-the-art results on 7 attributes among all the 12 facial attributes. For Duke attribute dataset, we achieve mean accuracy of 87.53% over 10 pedestrian attributes, which is 1.11% absolute percentage points higher than the best result reported so far (86.42%) by APR in [17] which, unlike ours, requires additional annotation information of person identification to assistant the learning pedestrian attributes. This further validates the efficacy of our adaptively weighted multi-task network. We also list the mean accuracy of different methods in Tab. 2.

(2) The efficacy of weighted multi-task network. Still as compared in Fig. 6, Fig. 7 and Tab. 2, we can draw the same conclusion as in CelebA dataset: our network is significantly better than the other two multi-task baselines. Particularly, on market-1501 dataset, our model shows 1.81% and 1.65% improvements over two multi-task baselines – Separate models and Basic model respectively. As expected, the same experimental results are observed on Duke attribute dataset: our results improve by 2.08% and 1.62% over Separate models and Basic model. Since the same basic networks are utilized in all methods, the performance gain is largely due to the efficacy of our validation loss trend algorithm and the loss layer.

Qualitative examples. The first two rows of Fig. 8 give some examples of the pedestrian attributes. Each column is corresponding to one type of pedestrian attribute. Fig. 9 compares the attribute prediction results of Separate models, Basic model and our model. The first row is ground truth attribute annotations of each image. As is shown in the table, the Separate models and Basic model made some incorrect predictions. For example, in the second image, the Basic model is missing the "Backpack" attribute, possibly due to the similar color of the backpack and clothes. In contrast, our model is more confidence in predicting the existence of "Backpack" attribute since our framework had learned the correlations among these attributes.

Figure 8: Person attributes. Image pairs in red/blue boxes indicate the positive/negative examples of representative attributes respectively.

Ground Truth	Hat, Male, Bag	Backpack, Male	Smiling, Hat Pointy Nose	Backpack, Female Shorts
Separate models	Hat, Male	Male	Hat, Pointy Nose	Backpack, Male, Shorts
Basic Model	Hat, Male	Male	Smiling, Hat	Female, Shorts
Our Model	Hat, Male, Bag	Backpack, Male	Smiling, Hat Pointy Nose	Backpack, Female Shorts

Figure 9: Examples of person attributes prediction result of separate models, basic model and our weighted model, respectively.

5 CONCLUSIONS

In this paper, we propose a novel Adaptively Weighted Multi-task Deep Convolutional Neural Network to learn person attributes. Different from previous multi-task approaches, our method utilizes validation loss as an indicator to adaptively tune weights for each attribute task. The proposed framework utilizes the correlation of all the attributes to help learn all the attributes tasks. We demonstrate our approach on the CelebA, Market-1501 attribute and Duke attribute datasets, showing substantial improvement over the state-of-the-art methods.

ACKNOWLEDGMENTS

The authors would like to thank Xuelin Qian for drawing Figure 2. This work was supported in part by the projects, #2017YFC0803702, #U1611461, #17511101702, and #ZN2016020103.

REFERENCES

[1] Abrar H Abdulnabi, Gang Wang, Jiwen Lu, and Kui Jia. 2015. Multi-task cnn model for attribute prediction. *IEEE TMM* (2015).

[2] Anna Bosch, Andrew Zisserman, and Xavier Munoz. 2007. Representing shape with a spatial pyramid kernel. In *ACM International Conference on Image and Video Retrieval*.

[3] Rich Caruana. 1998. Multitask learning. In *Learning to learn*. Springer, 95–133.

[4] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2014. Pedestrian Attribute Recognition At Far Distance. In *ACM MM*.

[5] Simon Denman, Clinton Fookes, Alina Bialkowski, and Sridha Sridharan. 2009. Soft-biometrics: Unconstrained Authentication in a Surveillance Environment. In *Digital Image Computing: Techniques and Applications*.

[6] Max Ehrlich, Timothy J Shields, Timur Almaev, and Mohamed R Amer. 2016. Facial attributes classification using multi-task representation learning. In *CVPR Workshops*.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385* (2015).

[8] J. Huang, R. S. Feris, Q. Chen, and S. Yan. 2015. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*.

[9] Rabia Jafri and Hamid R. Arabnia. 2009. A Survey of Face Recognition Techniques. In *Journal of Information Processing Systems*.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Gir-shick, S. Guadarrama, and T. Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv* (2014).

[11] B. Jou and S.F. Chang. 2016. Deep cross residual learning for multi-task visual recognition. In *ACM MM*.

[12] Brendan Jou and Shih-Fu Chang. 2016. Deep Cross Residual Learning for Multitask Visual Recognition. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 998–1007.

[13] Neeraj Kumar, Peter Belhumeur, and Shree Nayar. 2008. Facetracer: A search engine for large collections of images with faces. In *European conference on computer vision*. Springer, 340–353.

[14] Neeraj Kumar, Alexander Berg, Peter N Belhumeur, and Shree Nayar. 2011. Describable visual attributes for face verification and image search. *IEEE TPMAI* (2011).

[15] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. 2009. Attribute and simile classifiers for face verification. In *ICCV*. IEEE, 365–372.

[16] Giwoong Lee, Eunho Yang, and Sung Ju Hwang. 2016. Asymmetric Multi-task Learning Based on Task Relatedness and Loss. In *ICML*.

[17] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, and Yi Yang. 2017. Improving Person Re-identification by Attribute and Identity Learning. *arXiv preprint arXiv:1703.07220* (2017).

[18] L. Liu, J. Xing, S. Liu, H. Xu, X. Zhou, and S. Yan. 2014. Wow! you are so beautiful today! *ACM TMCCA* (2014).

[19] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. [n. d.]. Representation learning using multi-task deep neural networks for semantic classification and information retrieval.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *ICCV*.

[21] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. 2017. Fully-adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification. In *CVPR*.

[22] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[23] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. 2016. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249* (2016).

[24] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa. 2016. An All-In-One Convolutional Neural Network for Face Analysis. In *arxiv*.

[25] Ali Sharif Razavian, Josephine Sullivan, and Stefan Carlsson. 2014. CNN Features off-the-shelf : an Astounding Baseline for Recognition. *IEEE Conference on Computer Vision and Pattern Recognition'14 workshop on Deep vision* (2014).

[26] Ethan M. Rudd, Manuel Gunther, and Terrance E. Boult. 2016. MOON:A Mixed Objective Optimization Network for the Recognition of Facial Attributes. In *ECCV*.

[27] Behjat Siddiquie, Rogerio Feris, and Larry Davis. 2011. Image Ranking and Retrieval Based on Multi-Attribute Queries. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[28] S. Thrun. 1996. *Learning To Learn: Introduction.* Kluwer Academic Publishers.

[29] D.A. Vaquero, R.S. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. 2009. Attribute-based people search in surveillance environments. In *IEEE WACV*.

[30] Jing Wang, Yu Cheng, and Rogerio Schmidt Feris. 2016. Walk and Learn: Facial Attribute Representation Learning from Egocentric Video and Contextual Data. *CVPR*.

[31] S. Yang, P. Luo, C. C. Loy, and X. Tang. 2015. From Facial Parts Responses to Face Detection: A Deep Learning Approach. In *ICCV*.

[32] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.

[33] Kaipeng Zhang, Lianzhi Tan, Zhifeng Li, and Yu Qiao. 2016. Gender and smile classification using deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 34–38.

[34] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.

[35] Ning Zhang, Manohar Paluri, Marc'Aurelio Ranzato, Trevor Darrell, and Lubomir Bourdev. 2014. Panda: Pose aligned networks for deep attribute modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1637–1644.

[36] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2014. Facial landmark detection by deep multi-task learning. In *ECCV*.

[37] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for face alignment with auxiliary attributes. *IEEE transactions on pattern analysis and machine intelligence* 38, 5 (2016), 918–930.

[38] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. 2016. Learning deep representation for face alignment with auxiliary attributes. *IEEE TPAMI* (2016).

[39] R. Zhao, W. Ouyang, and X. Wang. 2014. Learning mid-level filters for person re-identification. In *CVPR*.

[40] L. Zheng, L. Shen, L. Tian, S.Wang, J.Wang, and Q. Tian. 2015. Scalable person re-identification: A benchmark. In *ICCV*.

[41] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017. Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in vitro. *arXiv preprint arXiv:1701.07717* (2017).

[42] Yang Zhong, Josephine Sullivan, and Haibo Li. 2016. Face Attribute Prediction Using Off-the-Shelf CNN Features. In *arxiv*.

[43] Qiang Zhou, Gang Wang, Kui Jia, and Qi Zhao. 2013. Learning to share latent tasks for action recognition. In *ICCV*.