Multi-Task CNN Model for Attribute Prediction

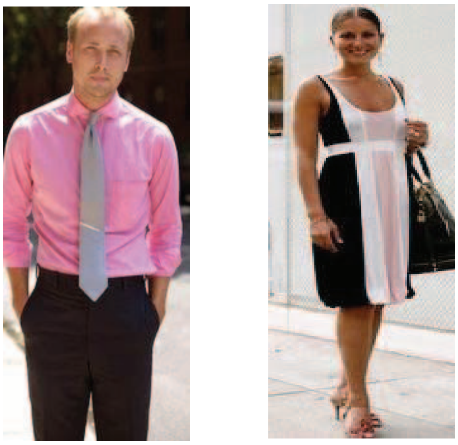
Abrar H. Abdulnabi, *Student Member, IEEE*, Gang Wang, *Member, IEEE*, Jiwen Lu, *Senior Member, IEEE*, and Kui Jia, *Member, IEEE*

***Abstract*—This paper proposes a joint multi-task learning algorithm to better predict attributes in images using deep convolutional neural networks (CNN). We consider learning binary semantic attributes through a multi-task CNN model, where each CNN will predict one binary attribute. The multi-task learning allows CNN models to simultaneously share visual knowledge among different attribute categories. Each CNN will generate attribute-specific feature representations, and then we apply multi-task learning on the features to predict their attributes. In our multi-task framework, we propose a method to decompose the overall model's parameters into a latent task matrix and combination matrix. Furthermore, under-sampled classifiers can leverage shared statistics from other classifiers to improve their performance. Natural grouping of attributes is applied such that attributes in the same group are encouraged to share more knowledge. Meanwhile, attributes in different groups will generally compete with each other, and consequently share less knowledge. We show the effectiveness of our method on two popular attribute datasets.**

***Index Terms*—Deep CNN, latent tasks matrix, multi-task learning, semantic attributes.**

I. Introduction

USING semantic properties, or attributes, to describe objects is a technique that has attracted much attention in visual recognition research [13], [30]. This is due to the fact that learning an object's attributes provides useful and detailed knowledge about it, and also serves as a bridge between low-level features and high-level categories. Various multimedia applications can benefit from attributes, among which are the following: knowledge transfer, information sharing between different target tasks, multimedia content analysis and recommendation, multimedia search and retrieval [5], [13], [14], [20], [30], [35], [36], [38], [43], [49].



Black: Yes, Necktie: Yes, Gender: Male, Strips: No, Yellow: No, White: No, Skin Exposure: No ...

Black: Yes, Necktie: No, Gender: Female, Strips: Yes, Yellow: No, White: Yes, Skin Exposure: Yes ...

Fig. 1. Illustration of binary semantic attributes. These examples are from the clothing attribute dataset [7]. Yes/No indicates the existence/absence of the corresponding attribute.

Typically, discriminative learning approaches are used to learn semantic attributes (attributes that have names)[13], [27], [30]. Fig. 1 shows two examples from the Clothing Attributes Dataset [7], where both images have different attribute labels. Other types of attributes, such as data-driven ones, are learned in an unsupervised or weakly supervised manner [32]. Relative attributes are also introduced and learned through ranking methods (relative attributes have real values to describe the strength of the attribute presence) [34], [45]. However, most of the existing literature employs discriminative classifiers independently to predict the semantic attributes of low-level image features [13], [30], [46]. Very few works model the relationship between object attributes, considering the fact that they may co-occur simultaneously in the same image [21], [44], [56].

Engineered low-level features like SIFT and HOG are used in combination with various loss-objective functions for attribute prediction purposes [13], [21], [30]. Improving the prediction results gives a good indication of the successful knowledge transfer of attributes between target tasks, for example, recognizing presently unseen classes through the transfer of attributes from another seen class [20], [30]. In the work of [13], attribute models are learned to generalize across different categories by training a naive Bayes classifier on the ground truth of semantic attributes. Then, they train linear SVM to learn non-semantic feature patterns and choose those which can be well predicted using the validation data attribute.

Manuscript received March 15, 2015; revised June 02, 2015; accepted August 15, 2015. Date of publication September 10, 2015; date of current version October 20, 2015. This work was supported in part by the Singapore Agency for Science, Technology, and Research (A*STAR) under a research grant for the Advanced Digital Sciences Center, in part by the Singapore Ministry of Education Tier 2 ARC28/14, and in part by the A*STAR Science and Engineering Research Council PSF1321202099. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Guo-Jun Qi.

A. H. Abdulnabi is with the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore 637553, and also with the Advanced Digital Sciences Center, Illinois at Singapore Pte. Ltd., Singapore 138632 (e-mail: abrarham001@ntu.edu.sg).

G. Wang is with the Department of Electrical and Electronic Engineering, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 637553 (e-mail: wanggang@ntu.edu.sg).

J. Lu is with the Department of Automation, Advanced Digital Sciences Center, Singapore 138632, and also with Tsinghua University, Beijing 100084, China (e-mail: jiwen.lu@adsc.com.sg).

K. Jia is with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China (e-mail: kuijia@umac.mo).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2015.2477680

The benefits of such generalization can be seen across different object categories, not just across instances within a category.

On the other hand, deep CNN demonstrates superior performance, dominating the top accuracy benchmarks in various vision problems [6], [17], [24], [50]. It has also been shown that CNN is able to generate robust and generic features [3], [40]. From a deep learning point of view, CNN learns image features from raw pixels through several convolutions, constructing a complicated, non-linear mapping between the input and the output. The lower convolution layers capture basic ordinary phrasing features (e.g., color blobs and edges), and the top layers are able to learn more complicated structure (e.g., car wheel)[59]. Subsequently, it is believed that such implementation of artificial Neural Networks mimics the visual cortex in the human brain [24].

Attribute prediction introduces two additional issues besides the typical object recognition challenges: image multi-labeling and correlation-based learning. Compared to single label classification, multi-labeling is more difficult. It is also more likely to appear in real word scenarios, at which point, an image would need to be tagged with a set of labels from a predefined pool [41], [54]. For example, when a user types their query to search for an image, such as ‘red velvet cake’, the engine should retrieve consistent results for real cake images having a red velvet appearance. This is a difficult task from a computational perspective due to the huge hypothesis space of attributes (e.g., M attributes required 2^M). This limits our ability to address the problem in its full form without transforming it into multiple single classification problems [57]. In particular, correlations should be explored between all these singular classifiers to allow appropriate sharing of visual knowledge. Multi-task learning is an effective method for feature sharing, as well as competition among classifiers (the so-called ‘tasks’ in the term multi-task)[21], [62]. If tasks are related and especially when one task lacks training data, it may receive visual knowledge from other, more fit tasks [1], [4], [19], [23], [31]. In the work of [21], they jointly learn groups of attribute models through multi-tasking using a typical logistic regression loss function.

Given the aforementioned issues, we propose an enhanced multi-task framework for an attribute prediction problem. We adapt deep CNN features as our feature representations to learn semantic attributes. Because the structure of the CNN network is huge, and thus requires powerful computation ability, we employ the following methods: First, if the number of attributes is small, we train multi-task CNN models together through MTL, where each CNN model is dedicated to learning one binary attribute. Second, if the number of attributes is relatively large, we fine-tune a CNN model separately on each attribute annotation to generate attribute-specific features, and then we apply our proposed MTL framework to jointly learn classifiers for predicting these binary attributes. The first approach is more or less applicable depending on the available resources (CPU/GPU and Memory). The visual knowledge of different attribute classes can be shared with all CNN models/classifiers to boost the performance of each individual model. Among the existing methods in multi-tasking, the work in [62] proposes a flexible method for feature selection by introducing a latent task matrix, where all categories are selected to share only the related visual knowledge through this latent matrix, which can also learn localized features. Meanwhile, the work in [21] interestingly utilizes the side information of semantic attribute relatedness. They used structured sparsity to encourage feature competition between groups and sharing within each of these groups. Unlike the work in [62], we introduce the natural grouping information and maintain the decomposition method to obtain the sharable latent task matrix and thus flexible global sharing and competition between groups through learning localized features. Also, unlike the work of [21], we have no mutual exclusive pre-assumptions, such that groups may not overlap with each other in terms of attribute members. However, as the hand-crafted feature extraction methods can limit the performance, we exploit deep CNN models to generate features that better suit the attribute prediction case.

We test our method on popular benchmarks attribute datasets: Animals with Attributes (AwA)[28] and the Clothing Attributes Dataset [7]. The results demonstrate the effectiveness of our method compared to standard methods. Because the Clothing dataset contains a small number of attributes, we successfully train our multi-task CNN model simultaneously. In addition because the AwA dataset contains a relatively large number of attributes, we first train each single CNN model on a target attribute. Then, we apply our multi-task framework on the generated features without instant back-propagation.

Our main contributions in this paper are summarized as follows. 1) We propose an enhanced deep CNN structure that allows different CNN models to share knowledge through multitasking. 2) We propose a new multi-task method; we naturally leverage the grouping information to encourage attributes in the same group to share feature statistics and discourage attributes in different groups to share knowledge. We relax any constraints on the groups, such as mutual exclusion, by decomposing the model parameters into a latent task matrix and a linear combination weight matrix. The latent task matrix can learn more localized feature, thus maintaining the ability to select some basic patterns through its configuration.

The remaining parts of our paper are summarized as follows: We first discuss the related work in Section II. The proposed method for the Multi-task CNN model in addition to the details of our MTL framework are presented in Section III. Experiments on two known attribute datasets and results are demonstrated in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

Because this work is mainly related to the topics of Semantic attributes, Multi-task learning and Deep CNN, we briefly review the most recent literature on these approaches including the following.

A. Semantic Attributes

Definition of Attribute: a visual property that appears or disappears in an image. If this property can be expressed in human language, we call it a Semantic property. Different properties may describe different image features such as colors, patterns, and shapes [13]. Some recent studies concentrate on how to link human-interaction applications through these mid-level attributes, where a consistent alignment should occur between human query expressions and the computer interpretations of query attribute phrases.

Global Versus Local Attributes: an attribute is global if it describes a holistic property in the image, e.g., 'middle aged' man. Usually, global attributes do not involve specific object parts or locations [21], [45], [60]. Localized attributes are used to describe a part or several locations of the object, e.g. 'striped donkey'. Both types are not easy to infer, because if the classifier is only trained on high-level labels without spatial information like bounded boxes, the performance of the under-sampled classifiers may degrade. However, some work in [21], [62] show that sharing visual knowledge can offset the effects of the lack of training samples.

Correlated Attributes: If attributes are related and cooccur they are correlated. In other words, some attributes will naturally imply others (e.g., 'green trees' and 'open sky' will imply 'natural scene'), so this configuration will impose some hierarchical relationship on these attribute classifiers. From another angle, attributes can be weaved from the same portion of the feature space and can be close to each other, e.g., 'black' and 'brown' attribute classifiers should be close to each other in the feature dimension space, belonging naturally to the same group, that is, the same color group [21]. While most of the existing methods train independent classifiers to predict attributes [13], [30], [55], typical statistical models, like naive Bayesian and structured SVM models, are used to address the problem. In [13], the authors employ a probabilistic generative model to classify attributes. In the work of [30], objects are categorized based on discriminative attribute representations. Some works flow by modeling the relationships between classes with pre-assumptions of existing attribute correlations [1]. Unlike this work, the decorrelation attribute method to resist the urge to share knowledge is proposed in [21], and they assume that attribute groups are mutually exclusive. Other work in [45] proposes jointly learning several ranking objective functions for relative attribute prediction.

Attributes and Multi-Labeling: Image multi-labeling is simply learning to assign multiple labels to an image [37], [54]. If the problem is adapted as is, a challenge arises when the number of labels increases and the potential output label combinations become intractable [54]. To mitigate this, a common transformation way is performed by splitting the problem into a set of single binary classifiers [42]. Predicting co-occurring attributes can be seen as multi-label learning. On the other hand, most of the related works [21], [45] tend to apply multi-task learning to allow sharing or using some label relationship heuristics a priori [11]. Another work applies ranking functions with deep CNN to rank label scores [17].

B. Multi-Task Learning

Why Mutli-Task Learning (MTL)? MTL has recently been applied to computer vision problems, particularly when some tasks are under-sampled [1], [4]. MTL is intended to impose knowledge sharing while solving multiple correlated tasks simultaneously. It has been demonstrated that this sharing can boost the performance of some or sometimes all of the tasks [4].

Task and Feature Correlations: Many strategies for sharing have been explored; the first one considers designing different approaches to discover the relationships between tasks [18], while the other considers an approach that aims to find some common feature structure shared by all tasks or mine the related features [39]. Recent frameworks, like Max-margin [61], Bayesian [58], and their joint extension [31], try to discover either or both task and feature correlations. While Max-margin is known by its discriminative power, Bayesian is more flexible and thus better suited to engage any a priori or performance inference [31]. In contrast to these studies, the work in [16] claims that as in typical cases, the dimension of the data is high; thus, the assumption that all the tasks should share a common set of features is not reasonable. They address such assumptions by simultaneously capturing the shared features among tasks and identifying outliers through introducing an outlier matrix [16].

In other works, [26], [62], the authors further relax the constraint naturally by decomposing the model parameters into a shared latent task matrix and linear combination matrix; hence, all the tasks are encouraged to select what to share through this latent matrix, which can learn more localized features. However, among all these techniques, they rely on popular ways to perform such sharing through applying various regularizations on the model parameters, such as structure sparsity for feature selection and feature competition [1].

C. Deep CNN

CNN for Feature Learning: CNN was born in the Deep Learning (DL) era [3], [6], [24]; its goal is to model high-level abstractions of visual data by using multiple non-linear transformation architectures. Among the DL models, CNN shows extraordinary performance, specifically in image classification and object recognition applications [6], [24]. Two bothersome issues about training CNN are the number of training samples needed and the time that is required to fully train the network. This means that to have an effective CNN model, the training dataset and time should be large enough for the CNN to gain the ability to perform its task well [24], [50]. The learned features generated by the CNN are shown to be robust, generic, and more effective than hand-crafted features [12], [40]. Fortunately, popular implementations of [12], [24] alongside the usage of pre-trained CNN models on the Imagenet dataset [24] make it easier to fine-tune various CNN architectures for many vision datasets.

CNN Between Single and Multi-Labels: Using CNN for single label prediction is intensively studied [3], [24]. There are many challenges that accompany multi-labeling, as previously discussed in the 'Attributes and Multi-labeling' section. Hence, training CNN directly is infeasible and impractical. However, one recent work proposes a work-around solution for the multi-label problem [57]. In this work [57], a shared CNN is fed with an arbitrary number of object segment hypotheses (image batches), which are extracted or generated by some techniques, like the binarized normed gradients (BING) [10]. The final CNN output results for all of these hypotheses are aggregated by max-pooling to give the final format of the multi-label predictions. Unlike their approach, our proposed

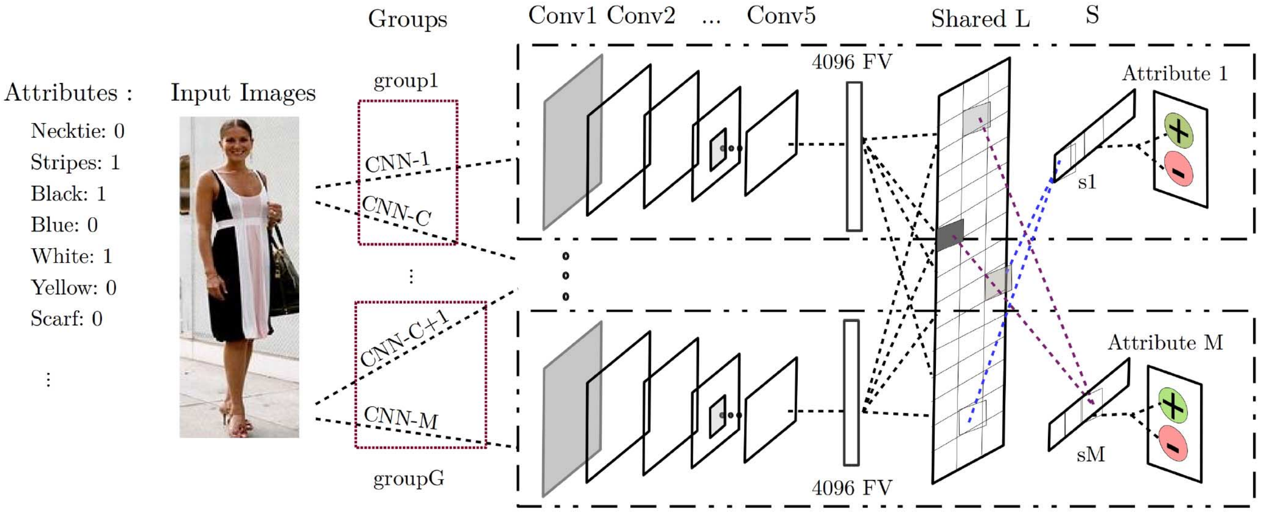


Fig. 2. Multi-task CNN models: the input image (left) with attribute labels information is fed into the model. Each CNN will predict one binary attribute. The shared layer L together with the S layer form a weight matrix of the last fully connected layer followed by a soft-max. The L layer is a shared latent matrix between all CNN models. Each vector in S is CNN-specific weight matrix layer. The soft-max and loss layers are replaced by our multi-task squared hinge loss. Group information about attribute relatedness is utilized during the training of the network.

model holds the essence of tagging one image with multiple labels through multi-task CNN models, which are simultaneously trained through MTL to allow sharing of the visual knowledge. Another direction for multi-labeling is proposed by [40], where the CNN is mainly used to generate off-the-shelf activation features; then, they apply SVM for later classifications. In our approach, when the number of attributes is large, we fine-tune many CNN models, each of which is dedicated to learning attribute-specific representations. These representations are used as off-the-shelf features for later stages in MTL, as we freeze their training while optimizing the multi-task loss function.

Convexity as First Aids for CNN: Some recent work [48], [51] demonstrates that convex optimization can improve the performance of highly non-convex CNN models. The authors in [48] propose modifying the last two layers in the CNN network by making a linear combination of many sub-models and then replacing the original loss function by other ones from the convex optimization family. One of their findings is that hinge loss is one of the preferable convex functions that performs well during backpropagation. Another work [51] confirms their finding that using the well-known SVM squared hinge loss does improve the final performance after training the CNN. By utilizing such experimental integration findings, we adopt a squared hinge loss framework to jointly optimize all classifier models while applying multi-tasking to naturally share visual knowledge between attribute groups.

In contrast to previous methods, our proposed approach is to train multi-task classifier models on deep features for attribute prediction and leverage a sharable latent task matrix that can be very informative for generating a full description of the input image in terms of attributes. Exploring the importance of such a latent matrix is a topic of future interest.

III. Multi-Task CNN Models

In this section, we will explain the details of the proposed approach of the multi-task CNN model. Fig. 2(c) shows the overall structure of the proposed method, starting from raw images and ending with attribute predictions. Given a vocabulary of M attributes, each CNN model will learn a binary nameable attribute. After the forward pass in all of the CNN models, the features generated from the last convolution layers will be fed into our joint MTL loss layer. To illustrate this more clearly, the weight parameter matrix learned in the loss layer will be decomposed into a latent task matrix and a combination matrix. The latent matrix can be seen as a shared layer between all the CNN models; in other words, the latent layer serves as a shared, fully-connected layer. Meanwhile, the combination matrix contains the specific information of each CNN model. It can also be seen as a specific fully-connected layer plugged above the shared latent fully-connected layer. After optimizing the joint loss layer and sharing the visual knowledge, each CNN model will take back its specific parameters through backpropagation in the backward pass. By presenting images that are annotated against several attributes, we iteratively train the whole structure until convergence.

We adopt the popular network structure proposed by Krizhevsky [24], which consists of 5 convolutions, followed by 2 fully-connected layers and finally the softmax and the loss. In addition, some pooling, normalization, and ReLU are applied between some of these layers. Many works have studied and analyzed the nature of this structure and identified some important aspects. For example, the work in [59] shows that the parameters of the fully-connected layers occupy almost 70% of the total network capacity, which consumes a great deal of effort while training the network. However, given the expense of trading between 'good-but-fast' and 'perfect-but-slow', the work in [3] shows that the performance will drop slightly when removing the fully-connected layers. Because our model requires more than one CNN model, we remove the last fully connected layers, as we substitute these layers with our own joint MTL objective loss, depending on the weight parameter matrix learned within.

In the following subsections, we demonstrate the shared latent task matrix (which also can be seen as a shared layer in the multi-task CNN models approach). Then, we show how the feature sharing and competition is engaged. Next, we introduce our formulations, which we use to solve the attribute prediction problem. Finally, the total optimization procedure used to train the whole network of multi-task CNN models is described. In the remaining part of the paper, we will use the task/CNN model as an interchangeable meaning for classifier because in all cases we employ the same underlying MTL framework. The only difference is that in one approach, the attribute-specific feature learning is on-line, and the MTL joint cost function optimization changes will affect the bottom layers in all CNN models through back-propagation. Thus, any shared knowledge will also be back-propagated to the bottom layers. Meanwhile, in the other approach, we learn these attribute-specific features in isolation of optimizing the joint cost function, as training many on-line CNN models on a large number of attributes is impractical.

A. Sharing the Latent Task Matrix in MTL

Given M semantic attributes, the goal is to learn a binary linear classifier for each of them. Each classifier or task has model parameters, which are denoted by w^m and are dedicated to predicting the corresponding attribute. W is the total classifier weights matrix, which can also be considered a softmax weights matrix but stacked from all CNN softmax layers. Given N training images, each of them has a label vector Y of $M - dimension$, such that Y^i_m is either $\{1\}$ or $\{-1\}$, indicating whether a specific training image i contains the corresponding m attribute having a value of $\{1\}$ or not $\{-1\}$. Suppose that the output from the last convolution layer in each CNN model forms our input feature vectors, such that each CNN model will generate an attribute-specific training pool. Thus, we will have X^N_M training examples aggregated from all CNN models.

Our assumption is inspired from the work of [62], where each classifier can be reconstructed from a number of shared latent tasks and a linear combination of these tasks. Through this decomposition, simultaneous CNN models can share similar visual patterns and perform flexible selection from the latent layer, which learns more localized features. We denote L to be this latent task matrix, and s^m is an attribute-specific linear combination column vector. In total, we have S linear combination matrices for all attribute classifiers.

Now, we want to split W into two matrices L and S, as we assume that W is a result of multiplying the shared L latent matrix and the combination matrix S, $W = LS$. To be more specific about each attribute classifier, the weight parameter vector can be formed by multiplying L with the corresponding s^m vector

$$w^m = Ls^m \tag{1}$$

where m is the index of the m-th attribute, $\{m = 1, 2, 3 \ldots M\}$.

Given the CNN models, we aim to learn the matrix W, which is formed by stacking the parameter matrices of the softmax layers of each CNN. The key idea behind our model is to decompose this weight matrix W into two matrices L and S, where the latent L matrix is the shared layer between all CNN models, S is a combination matrix, and each column corresponds to one CNN classification layer.

TABLE I
EXAMPLES OF ATTRIBUTE GROUPS FROM AWA DATASET [28]

Texture: patches, spots, stripes, furry, hairless, tough-skin
Shape: big, small, bulbous, lean
Colors: black, white, blue, brown, gray, orange, red, yellow
Character: fierce, timid, smart, group, solitary, nest-spot, domestic

TABLE II
GROUPING INFORMATION USED IN CLOTHING DATASET [7]

Group	**Attributes**
Colors	black, blue, brown, cyan, gray, green, many, red, purple, white, yellow
Patterns	floral, graphics, plaid, solid, stripe, spot
Cloth-parts	necktie, scarf, placket, collar
Appearance	skin-exposure, gender

By this decomposition, each CNN can share visual patterns with other CNN models through the latent matrix L, and all CNN models can collaborate together in the training stage to optimize this shared layer. Each CNN predicts whether the image contains the corresponding property. The benefit of learning the shared layer through multi-task is that each CNN can leverage the visual knowledge from learning other CNN models even if its training samples are not enough.

B. Feature Sharing and Competition in MTL

According to the semantic attribute grouping idea proposed in [21], group details are used as discriminative side information. It helps to promote which attribute classifiers are encouraged to share more visual knowledge, due to the group membership privileges. Meanwhile, different groups tend to compete with each other and share less knowledge. Table I shows some group information examples from the Animal with Attribute (AwA) dataset [28]. Because attributes are naturally grouped, we encode the grouping side information by encouraging attributes to share more if they belong to the same groups and compete with each other if they belong to different groups. Our attribute group information is shown in Table II.

Suppose we have M attributes and G groups, where each group contains a variable number of attributes, for example, g_1 contains $[a_1, a_2, \ldots, a_C]$ as shown in the left side of Fig. 2(c), and each group can have a maximum of M attributes. We have no restrictions on intra-group attributes. Even if two groups have the same attribute, the latent layer configuration mitigates the effect of the overlapped groups through the ability to learn more localized features. However, in our experiments, we rely on existing grouping information provided in the datasets, and obviously the groups are mutually exclusive (an attribute can be seen only in one group). Typically, solving the problem of overlapping groups requires some interior-point method, which is a type of second-order cone programming as discussed in [9], which is computationally expensive. Structured learning methods like group lasso [22] are applied in many areas employing such grouping information. Knowing any a priori information about the statistical information of features will definitely aid the classifiers. Hence, in our MTL framework, we utilize rich information of groups and also adopt a flexible decomposition to learn different localized features through the latent matrix. We follow the work in [21], as they also applied such group information of attributes.

Regularizations are our critical calibration keys to balance feature sharing of intra-group attribute classifiers and feature competition between inter-group attribute classifiers. The idea is that when applying the L_1 norm as $\sum_{m=1}^{M} ||w||_1$ [52], it will consequently encourage the sparsity on both rows/features and columns/tasks of W. The effect of sparsity on the rows will generate a competition scenario between tasks; meanwhile, the sparsity effect on the columns will generate sparse vectors. Additionally, when applying the L_{21} norm as $\sum_{d=1}^{D} ||w_d||_2$ [1], where D is the feature dimension space, in our case, because it is extracted from the previous layer, D is 4096. This can be seen as applying the L_1 norm on the zipped column-wise output of the L_{21}, which forces tasks to select only dimensions that are sharable by other tasks as a way to encourage feature sharing. As a middle solution [21], if the semantic group information is used when applying the L_{21} norm, the competition can be applied on the groups; meanwhile, the sharing can be applied inside each group.

In our framework, we encourage intra-group feature sharing and inter-group feature competition through adapting the L_{21} regularization term. Thus, we apply this on the vector set $s\sum_{k=1}^{K}\sum_{g}^{G} ||s_k^g||_2$ [1], [21], where K is the number of latent tasks (latent dimension space) and G is the number of groups, where each group contains a certain number of attributes. Specifically, s_k^g is a column vector corresponding to a specific attribute classification layer, and given a certain latent task dimension, s will contain all the intra-group attribute vector sets. This will encourage attribute classifiers to share specific pieces of the latent dimension space, if they only belong to the same group. Meanwhile, different groups will compete with each other as each of them tries to learn a specific portion from the latent dimension space. Additionally, the L_1 norm is applied on the latent matrix $||L||_1$ [52], [62], to learn more localized visual patterns.

C. Formulations of the Multi-Task CNN Model

Given the above discussions about decomposing W and by applying regularization alongside grouping information for better feature competition and sharing, we propose the following objective function:

$$\min_{L,S} \sum_{m=1}^{M}\sum_{i=1}^{N_m}\frac{1}{2}[max(0,1-Y_m^i(Ls^m)^T X_m^i)]^2 + \mu\sum_{k=1}^{K}\sum_{g=1}^{G}||s_k^g||_2 + \gamma||L||_1 + \lambda||L||_F^2. \quad (2)$$

This is the typical squared hinge loss function, in addition to our extra regularizations. For the m-th attribute category, we denote its model parameter as Ls^m and the corresponding training data is $(X_m^i, Y_m^i)_{i=1}^{N_m} \subset \mathbb{R}^d \times \{-1,+1\} (m = 1, 2, \ldots, M)$, where N_m is the number of training samples of the m-th attribute, and K is the total latent task dimension space. In the second term and given a specific latent task dimension k, s_k^g is a column vector that contains specific group attributes. The effect of this term is to continually elaborate on encouraging intra-group attributes to share feature dimensions. Thus, the columns/tasks in the combination matrix S will share with one another only if they belong to the same group. Such competition between groups is appreciated; however, if there is some overlap between groups (they are not absolutely disjointed), some mitigation may help through the latent matrix L configuration, which can learn more localized features. The L_1 norm is applied on the latent task matrix L to enforce sparsity between hidden tasks. The last term is the Frobenius norm to avoid overfitting. Moreover, with such a configuration of the latent matrix L, an implicit feature grouping is promoted. Namely, the latent tasks will allow finding a subset of the input feature dimensions D, which are useful for related tasks, where their corresponding parameters in the linear combination matrix S are nonzero.

Accordingly, every CNN is responsible for learning better input feature representations. Later in the testing, the input image will be fed into all CNN models to generate different input feature vectors; then the corresponding classifier weight vector will be applied to produce the attribute predications.

The bottom layers in each CNN model are defined in the same way as the network structure proposed by [24]. As shown in Fig. 2(c), every block of CNN has several hidden layers, mainly 5 convolutions. We replace the last 2 fully connected layers, softmax and the loss by our proposed MTL squared maxing hinge loss layer. Nevertheless, when the number of attributes is large, we freeze the training of the bottom layers and optimize the multi-task loss function to predict attributes, using the outputs generated from the CNN models.

D. Optimization Steps

Recall, that during the training procedure of M CNN models, each of them is responsible for predicting a single attribute. Our goal is to impose visual knowledge sharing between all CNN models through optimizing the multi-task objective function. The optimized components of W will serve as the last two fully connected layers. The L component is a shared layer between all CNN models. The generalization ability of each single CNN is improved by leveraging the shared visual knowledge from other attribute classifiers. The burden of the Stochastic Gradient Descent (SGD) optimizer is only centralized in terms of training the bottom layers well if they are not freeze from training, so that each CNN can provide robust feature representation of images.

Solving the proposed cost function is non-trivial, because it is not jointly convex on either L or S. The work in [44] solves the non-convex regularized function using the block coordinate descent method. The function hence becomes a bi-form convex function. They employ Accelerated Proximal Gradient Descent (APG) to optimize both L and S in an alternating manner. Specifically, if S is fixed, the function becomes convex over L, and optimizing it by APG can solve this state and handle the non-smooth effect of the l_1 norm. Likewise, if L is fixed, the function becomes convex over S; in this form of the function and unlike [44], the mixed norm regularizations require re-representing the 2-norm into its dual form as discussed in [21]. Smoothing Proximal Gradient Descent (SPGD) [9], [21], [22] is applied to obtain the optimal solution of S. These optimizations are common in the literature of structured learning, where various regularizations may disturb convexity and smoothness properties of the functions. Algorithm III-D illustrates the main steps that are applied to optimize (2).

Furthermore, when L is fixed, the optimization problem is in terms of S and is described as follows:

$$\min_{L,S} \sum_{m=1}^{M} \sum_{i=1}^{N_m} \frac{1}{2} [max(0, 1 - Y_{m^i} (Ls^m)^T X_m^i)]^2 + \mu \sum_{k=1}^{K} \sum_{g=1}^{G} ||s_k^g||_2. \quad (3)$$

Optimization by SPGD: Chen *et al.* [9] propose solving optimization problems that have a mixed-norm penalty over a priori grouping to achieve some form of structure sparsity. The idea behind this optimization is to introduce the smooth approximation of the objective loss function and solve this approximation function instead of optimizing the original objective function directly. Some work proposes solving non-overlapping groups, as is the case in [21]. Others extend the solution to overlapping groups, as in [9]. We closely follow the approach of approximating the objective function proposed in tree-guided group lasso [22], [47], which is basically built on the popular group-lasso penalty [2]. We apply the step of squaring the mixed-norm term [21], which is originally suggested in [2]. Squaring before optimization makes the regularizer positive, which generates a smooth monotonic mapping, preserving the same path of solutions but making the optimization easier. For further details on this approximation, refer to [2].

Now, after fixing S, the optimization problem is in terms of L as follows:

$$\min_{L,S} \sum_{m=1}^{M} \sum_{i=1}^{N_m} \frac{1}{2} [max(0, 1 - Y_m^i (Ls^m)^T X_m^i)]^2 + \gamma ||L||_1 + \lambda ||L||_{F^2}. \quad (4)$$

Optimization by APG: Accelerated Proximal Method updates the searching point from the last linear combination of two points in each iteration and thus converges faster [53]. Furthermore, it also handles non-smooth convex functions using proximal operators. The idea is to rely on a shrinkage operator [44], [53] while updating the search point given the previous one and the gradient of the smooth part of the function (the non-regularized part). We adopt this method to optimize over L because the proximity operator is straightforward, as the non-smooth l_1 norm has been studied extensively [2], [52]. Meanwhile, in other general learning problems, the proximity operator cannot be computed explicitly, namely, the mixed-norm regularization term; hence, we adopt SPGD while optimizing S. We can optimize L through SPGD by using approximations for both the gradient and the proximity operator; however, the APG has a relatively lower convergence rate.

Algorithm 1: Solving the Optimization Problem of (2)

```
Input — Generated features from CNN models: $X_M^N$
        Attributes labels with values $\{-1, 1\}$: $Y_M^N$
Output — Combination weight matrix $S$
         Latent tasks matrix $L$
         Overall Model weight matrix $W$
Step 1 — Fix $L$ and optimize $S$ by SPGD
         Solving (3) until convergence
Step 2 — Get $S$ from Step 1, and optimize $L$ by APG
         Solving (4) until convergence
Step 3 — Repeat Step 1 and Step 2
         Solving (2) until convergence
```

During a training epoch, the forward pass will generate the input for the multi-task loss layer from all the CNN models. After optimizing (2) using the proposed algorithm III-D, the output is the overall model weight matrix W, where each column in W will be dedicated to its specific corresponding CNN model and is taken back in the backward pass alongside the gradients with respect to its input. W is reconstructed using the optimal solutions of L and S, where knowledge sharing is already explored through MTL between all the CNN models via L.

IV. Experiments and Results

A. Datasets and Grouping

We conduct our experiments on two datasets:

Clothing Attributes Dataset: This dataset is collected by the work in [7]; it contains 1856 images and 23 binary attributes, as well as 3 multi-class value attributes. The ground-truth is provided on image-level, and each image is annotated against all the attributes. We ignore the multi-class value attributes, because we are only interested in binary attributes. The purpose behind such clothing attributes is to provide better clothing recognition. We train Multi-task CNN models to predict the attributes in this dataset. Because no grouping information is suggested in this dataset, we follow the natural grouping sense proposed in other attribute datasets as in [28]. In Table II, we show the details of our attribute grouping on the Clothing Attributes dataset.

AwA Dataset: The Animals with Attributes dataset is collected in [28], the purpose of which is to apply transfer learning and zero-shot recognition [29]. It consists of 30475 images of 50 animal classes. The class/attribute matrix is provided; hence, the annotation is on the animal's class level. It provides 85 binary attributes for each class. This dataset has 9 groups: colors, textures, shapes, activity, behavior, nutrition, character, habitat and body parts [21]; Table I shows some attributes in some of these groups.

B. Attribute Prediction Accuracy

We conduct several experiments on these two datasets. For the clothing dataset, we train multiple CNN models simultaneously. We calculate the accuracy of attribute predictions against

TABLE III
ACCURACY OF ATTRIBUTE PREDICTION BEFORE SHARING THE L LAYER, AFTER SHARING AND PREVIOUS METHODS ON THE CLOTHING DATASET [8]. G1 REFERS TO COLOR ATTRIBUTES, G2 REFERS TO THE PATTERN GROUP, G3 REFERS TO CLOTH-PARTS, AND G4 REFERS TO THE APPEARANCE GROUP. MG-CNN IS OUR OVERALL PROPOSED FRAMEWORK. THE HIGHER, THE BETTER. FOR FURTHER DETAILS ABOUT SEVERAL SITTING AND METHOD NAMES IN THIS TABLE, REFER TO SECTION IV-B

Method	G1	G2	G3	G4	Total
S-extract	81.84	82.07	67.51	69.25	78.31
M-extract	84.98	89.89	81.41	81.03	85.29
S-CNN	90.50	92.90	87.00	89.57	90.43
M-CNN	91.72	94.26	87.96	91.51	91.70
MG-CNN	93.12	95.37	88.65	91.93	**92.82**
CF [15]	81.00	82.08	77.63	78.50	80.48
CRF [7]	85.00	84.33	81.25	82.50	83.95

TABLE IV
ATTRIBUTE DETECTION SCORES OF OUR MULTI-TASK FRAMEWORK COMPARED WITH OTHER METHODS ON AWA [28]. THE HIGHER, THE BETTER. (MEAN AVERAGE PRECISION)

Tasks	Prediction Score
lasso [52]	61.75
l_{21} all-sharing [1]	60.21
l_2 regression loss	66.87
decorrelated [21]	64.80
category-trained CNN [33]	74.89
single CNN	75.37
multi-task CNN (ours)	**81.19**

the provided ground truth in the dataset. In Table III, S-extract refers to a simple sitting where we directly use a pre-trained model of CNN [25] for feature extraction, and then we train single SVM tasks for attribute prediction; meanwhile, in M-extract, we train our MTL framework on the same CNN extracted features. S-CNN refers to the single-task CNN, where we fine-tuned individual models of CNN to predict each attribute, and M-CNN refers to our MTL framework without encoding the group information [62], and MG-CNN is our whole MTL framework with group encodings and wholly training CNN models with our framework together. CF refers to the combined features model with no pose baseline [7], while CRF refers to the state-of-the-art method proposed by [7]. Our model outperforms the state-of-the-art results in [7]. We notice, though, that the overall improvement margin over the single CNN task models is relatively small compared to our results in AwA (see Table IV). This is because the accuracy results are already quite high and thus hard to improve further.

We conduct another experiment on the AwA dataset. We fine-tuned single CNN models separately on each attribute. Later, given the input images, we use these learned models to extract attribute-specific features. In other words, we freeze the training of the bottom layers in all CNN models and elaborate only in training our multi-task loss layer. This is due to the large number of attributes in the AwA dataset. We note that the fine-tuning stage will not add much practical difference and is a very time consuming process, perhaps due to the fact that AwA and Image-net datasets have an overlap of approximately 17 object categories; this has also been explored by another work [33], in which they even train the CNN model to classify objects on the AwA dataset; however, they noticed that using the pre-trained CNN model on the Imagenet dataset directly or fine-tuning the model on AwA will in both cases give the same attribute prediction results. However, our MTL framework outperforms the single-tasks by a large margin; Table IV shows the performance of our method compared with other standard methods, where the prediction accuracy is in terms of the mean average over all of the attributes. Compared with previous state-of-the-art results, which are approximately 75% [33], our trained MTL CNN models on attributes outperform it by a large margin. Additionally, Table V shows the accuracy results in terms of mean average precision over each group of attributes (group-level accuracy), before and after applying our multi-task framework. Initialization of the pre-trained model on Imagenet [24] again is used throughout our experiments. Fig. 3 also shows a number of misclassified test samples from single-task classifiers, which our multi-task classifiers classified correctly.

TABLE V
GROUP-LEVEL ACCURACY RESULTS OF OUR MULTI-TASK FRAMEWORK ON AWA [28]. THE HIGHER, THE BETTER

Groups	# Attributes	Single CNN	Our Multi-task CNN
Colors	9	76.91	82.28
Texture	6	76.16	82.44
Shape	4	61.67	72.68
Body-Parts	18	75.93	81.82
Activity	10	82.22	85.4
Behavior	5	72.78	74.96
Nutrition	12	74.76	82.67
Habitat	15	80.21	84.72
Character	7	62.01	70.52
Total	**85**	**75.37**	**81.19**

C. Implementation Details

CNN Model Training: We put each CNN model through 100 epochs, although most models converged in approximately 50 epochs. We initialize each CNN model with the pre-trained network on Imagenet [24] and fine-tune it on the target attribute annotations. We normalize the input images into 256 × 256 and subtract the mean from them. To train our model on the Clothing dataset, we use a data augmentation strategy as in [24].

Multi-Task Optimization: We perform Singular Value Decomposition (SVD) on W following the work in [62] to obtain an initialization for L; meanwhile, S is randomly initialized. The initialization of W was accomplished by stacking the last fully-connected layers from all pre-trained CNN models. The other model parameter values are either selected experimentally or following the typical heuristics and strategies proposed in [24]. The latent tasks number is set to the maximum possible feature dimension, because the application of attribute prediction is very critical for any subtle fine-grained details; thus, any severe information loss caused via SVD can degrade the performance drastically. Hence, the number of latent tasks is set to 2048 in our experiments.

Additionally, we set the weight decay to 0.0005 in our CNN models; also, the momentum is set to 0.9, and the learning rate is initialized by 0.01 and reduced manually throughout training; we follow the same heuristic in [25]. In our multi-task part [see (2)], the latent task $\lambda||L||_F^2$ regularization parameter λ is set to 0.4, and the other two parameters γ and μ are best validated in each dataset experiments with held out unseen attribute data.

We conduct our experiments on two NVIDIA TK40 16 GB GPU; the overall training time including the CNN part and MTL

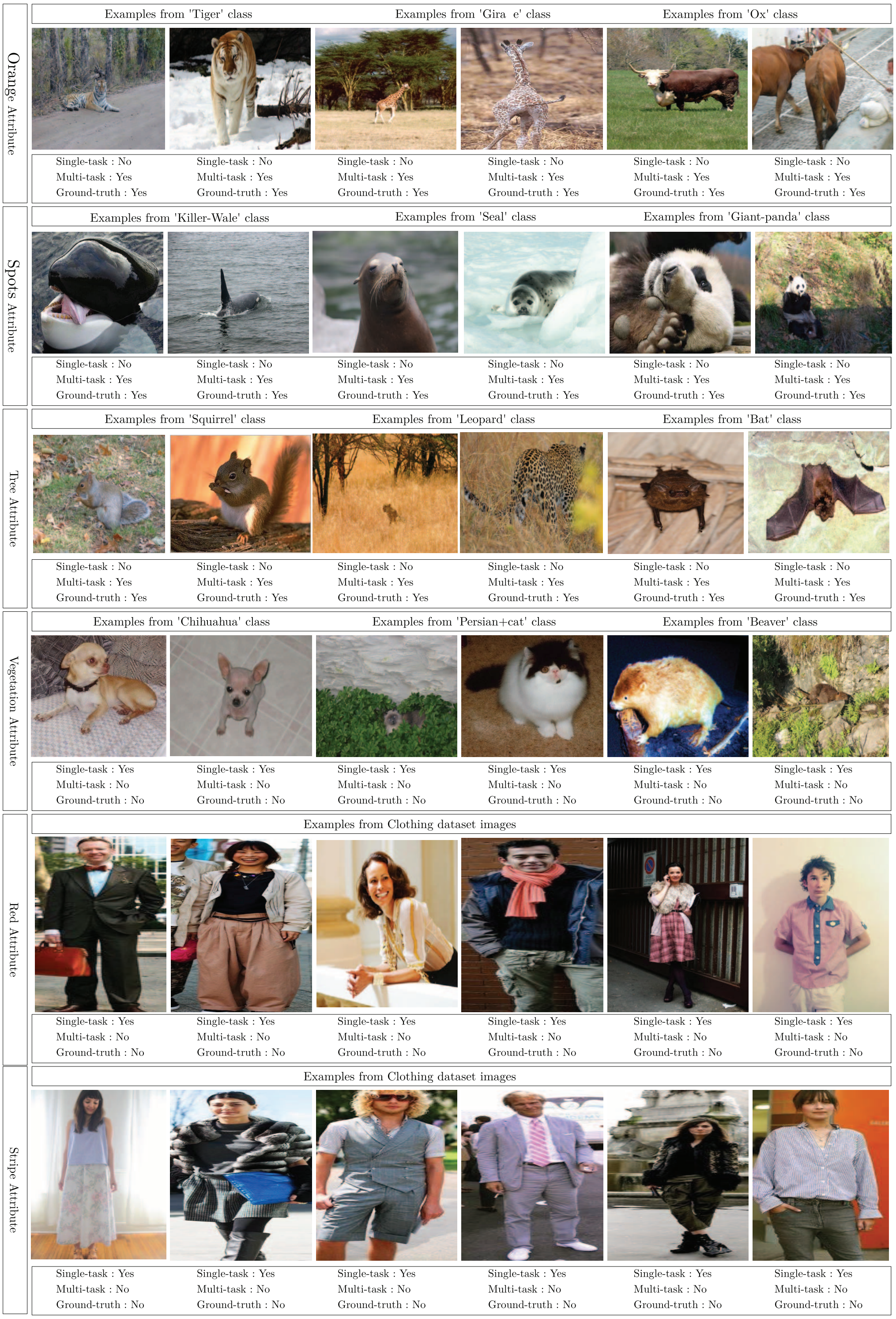


Fig. 3. Examples of misclassified samples from single-task classifiers results. The first four rows have samples from the AwA dataset [28]. The last two rows are samples from the clothing attributes dataset [7]. Yes/No indicates whether the attribute is presented or absent in the image. Multi-task classifiers are able to correctly classified these samples.

part of the training is approximately 1.5 days for the Clothing dataset (approximately 50 epochs for all 23 CNN models), and the testing time including feature extraction from all CNN models is approximately 50 minutes (sequential extraction from models one by one, not in parallel, where the time needed to extract features in each model is about 1.5 minutes/1000 images); if more attribute CNN models are added, the time will eventually increase. For the AwA dataset, we divide its training images into several sets (each set contains 3000 images, and we have 8 sets; 5 are used for training and 3 for testing). In total, the training time takes approximately 2 weeks (however, because we noticed that the major accuracy increase was mainly from training our MLT framework and not from CNN fine-tuning, we re-conducted the experiment and froze the bottom layers and depended on training the MTL layer, as we previously discussed; but in the second experiment, we saved a great deal of training time, as it only takes approximately 13 hours to completely train the last two layers on all CNN models within our MTL framework on one training set; on all remaining sets, it takes approximately 2.5 days to complete training.

V. Conclusion

In this paper, we introduce an enhanced multi-task learning method to better predict semantic binary attributes. We propose a multi-task CNN model to allow sharing of visual knowledge between tasks. We encode semantic group information in our MTL framework to encourage more sharing between attributes in the same group. We also propose decomposing the model parameters into a latent task matrix and a linear combination matrix. The latent task matrix can effectively learn localized feature patterns, and any under-sampled classifier will generalize better through leveraging this sharable latent layer. The importance of such a latent task matrix is a topic of future interest. Specifically, we would like to explore the potential of the latent task matrix decomposition to be informative enough to generate an efficient description of the input image in terms of either semantic or latent attributes. Our experiments on both attribute benchmark datasets show that our learned multi-task CNN classifiers easily outperform the previous single-task classifiers.

Acknowledgment

The authors would like to thank NVIDIA Corporation for their donation of Tesla K40 GPUs used in this research at the Rapid-Rich Object Search Lab. This research was carried out at both the Advanced Digital Sciences Center (ADSC), Illinois at Singapore Pte Ltd., Singapore, and at the Rapid-Rich Object Search (ROSE) Laboratory, Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, under its Interactive & Digital Media (IDM) Strategic Research Programme.

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proc. NIPS*, 2007, pp. 41–48.

[2] F. Bach, "Consistency of the group lasso and multiple kernel learning," *CoRR*, 2007 [Online]. Available: http://arxiv.org/abs/0707.3390

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[4] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[5] S. Chang, G.-J. Qi, J. Tang, Q. Tian, Y. Rui, and T. S. Huang, "Multimedia LEGO: Learning structured model by probabilistic logic ontology tree," in *Proc. 13th IEEE Int. Conf. Data Mining*, Dec. 2013, pp. 979–984.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC*, 2014.

[7] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," in *Proc. 12th Eur. Conf. Comput. Vis.—Vol. Part III*, 2012, pp. 609–623.

[8] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," in *Proc. ECCV*, 2012, pp. 609–623.

[9] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, "Smoothing proximal gradient method for general structured sparse learning," in *Proc. 27th Conf. Uncertainty Artificial Intell.*, 2012, pp. 1–10.

[10] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 3286–3293.

[11] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *Computer Vision—ECCV 2014* , ser. Lecture Notes in Comput. Sci., D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. New York, NY, USA: Springer, 2014, vol. 8689, pp. 48–64.

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. ICML*, 2013, pp. 647–655.

[13] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 1778–1785.

[14] V. Ferrari and A. Zisserman, "Learning visual attributes," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2007, pp. 433–440.

[15] A. Gallagher and T. Chen, "Clothing cosegmentation for recognizing people," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.

[16] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 895–903.

[17] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," *CoRR*, 2013 [Online]. Available: http://arxiv.org/abs/1312.4894

[18] B. Hariharan, L. Zelnik-manor, S. V. N. Vishwanathan, and M. Varma, "Large scale max-margin multi-label classification with priors," in *Proc. Int. Conf. Machine Learn.*, 2010, pp. 423–430.

[19] S. J. Hwang, F. Sha, and K. Grauman, "Sharing features between objects and their attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 1761–1768.

[20] D. Jayaraman and K. Grauman, "Zero shot recognition with unreliable attributes," *CoRR*, 2014 [Online]. Available: http://arxiv.org/abs/1409.4327

[21] D. Jayaraman, F. Sha, and K. Grauman, "Decorrelating semantic visual attributes by resisting the urge to share," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 1629–1636.

[22] S. Kim and E. Xing, "Tree-guided group lasso for multi-task regression with structured sparse," in *Proc. ICML*, 2010, pp. 543–550.

[23] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *Proc. ICML*, 2010, pp. 543–550.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Assoc., 2012, pp. 1097–1105.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[26] A. Kumar and H. Daume, "Learning task grouping and overlap in multi-task learning," in *Proc. 29th Int. Conf. Machine Learn.*, 2012, pp. 1383–1390.

[27] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc.12th IEEE Int. Conf. Comput. Vis.*, Sep.–Oct. 2009, pp. 365–372.

[28] C. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 951–958.

[29] C. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, Mar. 2014.

[30] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 951–958.

[31] C. Li, J. Zhu, and J. Chen, "Bayesian max-margin multi-task learning with data augmentation," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 415–423.

[32] S. Ma, S. Sclaroff, and N. Ikizler-Cinbis, "Unsupervised learning of discriminative relative visual attributes," in *Proc. ECCV Workshops*, 2012, vol. 12, no. 3, pp. 61–70.

[33] M. Ozeki and T. Okatani, "Understanding convolutional neural networks in terms of category-level attributes," in *Proc. ACCV*, 2014, pp. 362–375.

[34] D. Parikh and K. Grauman, "Relative attributes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 503–510.

[35] G.-J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. Huang, "Exploring context and content links in social media: A latent space method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 850–862, May 2012.

[36] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 17–26.

[37] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang, "Two-dimensional active learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.

[38] G.-J. Qi, X.-S. Hua, and H.-J. Zhang, "Learning semantic distance from community-tagged media collection," in *Proc. ACM Int. Conf. Multimedia*, 2009, pp. 243–252.

[39] P. Rai and H. D. Iii, "Infinite predictor subspace models for multitask learning," in *Proc. 13th Int. Conf. Artificial Intell. Statist.*, 2010, pp. 613–620.

[40] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2014, pp. 512–519.

[41] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases: Part II*, 2009, pp. 254–269.

[42] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Eur. Conf. Mach. Learn. Knowledge Discovery Databases: Part II*, 2009, pp. 254–269.

[43] O. Russakovsky and L. Fei-fei, "Attribute learning in large-scale datasets," in *Trends and Topics in Computer Vision*, ser. Lecture Notes in Comput. Sci.. Berlin, Germany: Springer-Verlag, pp. 1–14.

[44] R. Salakhutdinov, A. Torralba, and J. Tenenbaum, "Learning to share visual appearance for multiclass object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 1481–1488.

[45] R. Sandeep, Y. Verma, and C. Jawahar, "Relative parts: Distinctive parts for learning relative attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 3614–3621.

[46] W. Scheirer, N. Kumar, P. Belhumeur, and T. Boult, "Multi-attribute spaces: Calibration for attribute fusion and similarity search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2933–2940.

[47] E. P. X. Seyoung Kim, "Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping," *Ann. Appl. Stat.*, vol. 6, no. 3, pp. 1095–1117, 2012.

[48] Y. W. Si Chen, "Convolutional neural network and convex optimization," Dept. of Elect. and Comput. Eng., Univ. of California at San Diego, San Diego, CA, USA, Tech. Rep., 2014 [Online]. Available: http://acsweb.ucsd.edu/~yuw176/report/ECE273.pdf

[49] B. Siddiquie, R. Feris, and L. Davis, "Image ranking and retrieval based on multi-attribute queries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 801–808.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014 [Online]. Available: http://arxiv.org/abs/1409.1556

[51] Y. Tang, "Deep learning using linear support vector machines," *CoRR*, 2015 [Online]. Available: http://arxiv.org/abs/1306.0239

[52] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc.*, vol. 58, pp. 267–288, 1996.

[53] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *SIAM J. Optimization*, submitted for publication.

[54] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 2007, pp. 1–13, 2007.

[55] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. Blaschko, D. Weiss, B. Taskar, K. Simonyan, N. Saphra, and S. Mohamed, "Understanding objects in detail with fine-grained attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 3622–3629.

[56] G. Wang and D. Forsyth, "Joint learning of visual attributes, object classes and visual saliency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Sep.–Oct. 2009, pp. 537–544.

[57] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "CNN: Single-label to multi-label," *CoRR*, 2014 [Online]. Available: http://arxiv.org/abs/1406.5726

[58] M. Yang, Y. Li, and Z. University, "Multi-task learning with Gaussian matrix generalized inverse Gaussian model," in *Proc. 30th Int. Conf. Mach. Learn.*, May 2013, vol. 28, pp. 423–431.

[59] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

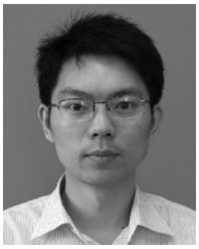
[60] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. D. Bourdev, "PANDA: Pose aligned networks for deep attribute modeling," *CoRR*, 2013 [Online]. Available: http://arxiv.org/abs/1311.5591

[61] Y. Zhang and J. Schneider, "Learning multiple tasks with a sparse matrix-normal penalty," in *Proc. NIPS*, 2010, pp. 2550–2558.

[62] Q. Zhou, G. Wang, K. Jia, and Q. Zhao, "Learning to share latent tasks for action recognition," in *Proc. ICCV*, 2013, pp. 2264–2271.

Abrar H. Abdulnabi (S'15) received the B.Eng. degree in computer engineering from the Jordan University of Science and Technology, Irbid, Jordan, in 2013, and is currently working toward the Ph.D. degree in electrical and electronic engineering at Nanyang Technological University, Singapore.

She is with the Advanced Digital Sciences Center, Illinois at Singapore Pte. Ltd., Singapore, and with the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore, under A*STAR's Singapore International Graduate Award (SINGA) scholarship. Her current research interests include computer vision, deep feature learning, pattern recognition, and machine learning.



Gang Wang (M'11) received the B.S. degree from the Harbin Institute of Technology in Electrical Engineering, Harbin, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Champaign, IL, USA, in 2010.

He is an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include computer vision and machine learning, particularly object recognition, scene analysis, and deep learning.

Dr. Wang is an Associate Editor of *Neurocomputing*. He was a recipient of the Harriett and Robert Perry Fellowship (2009–2010) and CS/AI Award (2009) at UIUC.

Jiwen Lu (S'10–M'11–SM'15) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from the Nanyang Technological University, Singapore, in 2011.

He is with the Department of Automation, Tsinghua University, Beijing, China. He has authored or coauthored over 110 scientific papers in these areas, with more than 40 papers published in IEEE transactions, journals, and top-tier computer vision conferences. His current research interests include computer vision, pattern recognition, machine learning, and multimedia computing.

Dr. Lu is currently serving as an Associate Editor of *Pattern Recognition Letters* and *Neurocomputing*. He was a recipient of the First-Prize National Scholarship and the National Outstanding Student Award from the Ministry of Education of China in 2002 and 2003, the Best Student Paper Award from Pattern Recognition and Machine Intelligence Association of Singapore in 2012, the Top 10% Best Paper Award from the IEEE International Workshop on Multimedia Signal Processing in 2014, and the National 1000 Young Talents Plan Program in 2015.

Kui Jia (M'12) received the B.Eng. degree in marine engineering from the Northwestern Polytechnic University, Xi'an, China, in 2001, the M.Eng. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2003, and the Ph.D. degree in computer science from the Queen Mary University of London, London, U.K., in 2007.

He is currently a Visiting Assistant Professor with the University of Macau, Macau, China. His research interests include computer vision, machine learning, and image processing.