MAJOR-SUBORDINATE-TASK LEARNING FOR IMAGE ORIENTATION ESTIMATION

Yilin He, Wengang Zhou, Houqiang Li

CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System,
EEIS Department, University of Science and Technology of China
heyilin@mail.ustc.edu.cn, {zhwg,lihq}@ustc.edu.cn

ABSTRACT

In this work, we propose a major-subordinate-task learning framework to estimate image orientation. The involved two tasks, regression to the characteristic orientation of the image (major) and classification by visual content (subordinate), are fed with shared feature and update feature extractor collaboratively. To boost the major task, we introduce a novel module, matched gradients weight multiplier, to calculate matching degree of the two tasks and adaptively adjust feedback from the subordinate task towards the shared feature extractor accordingly. As a result, such feedback is expected to be always promotive to the major task. Experiments demonstrate the effectiveness of our proposed framework over the counterpart settings.

Index Terms— Image orientation estimation, deep learning framework

1. INTRODUCTION

In recent years, Convolutional Neural Networks [1] (CNNs) have been widely used in various machine learning applications. Adoption of such deep architecture has resulted in state-of-the-art performances in many computer vision tasks including classification [2], semantic segmentation [3], detection [4] and image retrieval [5] [6]. Along with unprecedented modeling capability it exhibited in such tasks, some limitations remain unsolved, among which is the lack of ability to deal with rotation invariance simply by convolution and local pooling.

One intuitive solution to this limitation is data augmentation by involving training samples of various angles, which, however, suffers extremely high complexity in computation. Another promising alternative is data preprocessing: images are aligned to appropriate orientations before been fed to CNNs as input data. In this work, we follow the second direction. In the proposed method, a deep architecture is adopted for the estimation task considering their accuracy and computational efficiency displayed in an array of successful applications. Besides, discovery of a previous work [7] demonstrated that prediction accuracy of image orientation is largely dependent on its visual content, which inspired us to take images visual content into consideration while estimating its orientation. These lead to the formulation of our proposed learning framework. Two tasks are set in the framework, including a major task, *i.e.*, orientation estimation and a subordinate task, *i.e.*, content-relative classification. In contrast to equality of tasks in an ordinary multi-task learning framework, subordinate task in our proposed framework serves to promote performance of the major task while regardless of its own performance.

The general mechanism for our method is as follows. Two tasks are trained simultaneously forwarded with shared convolutional feature, while updates to shared feature extractor by the subordinate task is multiplied by a scalar, the matching confidence score, which is calculated stepwise in training stage. A novel module, matched gradients weight multiplier, is designed to sustain the calculation-multiplication while running standard stochastic gradient descent algorithms. This mechanism gives rise to subordinate task's persistent promotion for the major task, regardless of how well two tasks are matched.

The rest of this work is organized as follows. We first make a survey of related work in Section 2. After that, we elaborate our framework in Section 3 and discuss the experimental results in Section 4. Finally, we conclude this work in Section 5.

2. RELATED WORK

There are large bodies of works on orientation estimation published over the years. Traditional hand-crafted methods such as SIFT [8], ORB [9], BRISK [10] and FREAK [11] tend to divide the image into small patches and extract descriptors from each patch individually, which can well represent dominant orientation of key points, but is demonstrated infeasible to predict image's global orientation with these local contexts. Deep orientation predictors such as Spatial Transformer Networks [12] (STN) learns affine transformation parameters di-

This work was supported in part to Dr. Houqiang Li by 973 Program under contract No. 2015CB351803 and NSFC under contract No. 61390514, and in part to Dr. Wengang Zhou by NSFC under contract No. 61472378 and No. 61632019, the Fundamental Research Funds for the Central Universities, and Young Elite Scientists Sponsorship Program By CAST (2016QNRC001).

978-1-5386-1737-3/18/$31.00 ©2018 IEEE

rectly from the entire image with classification information as ground truth, then forwards the transformation which can align the feature map to the subsequent layer by differentiable bilinear interpolation. Since the geometric transformation parameters are learned without explicit supervision, it cannot guarantee to identify the genuine solutions as expected. Suggested by STN, we adopt a deep architecture for our orientation estimator but give every training sample a canonical orientation as ground truth. We show in Section 4 our experiment results to justify the robustness of our method in terms of images with arbitrary orientation.

Related are also numerous applications with multi-task learning [13] framework, which incarnated our idea to connect image's visual content with orientation estimation. Among the multi-task learning applications, works on joint optimization [14] [15] [16] take up a large portion. Works as [14] [15] addressed the idea that training in parallel with related tasks, learning for individual task will be improved since underlying correlation between semantic features, or "attributes" that are desirable for respective tasks are well harnessed. When relationship of two tasks is somewhat opposite to that in joint optimization, introducing adversarial training in one task brings advance to the other task because learned features are exclusively beneficial for the latter. This learning strategy is described in the work *Unsupervised Domain Adaptation by Backpropagation* [17].

Both approaches are meaningful and demonstrate to be effective in their own contexts, while not promising when the contexts are altered. Apparently, either training poorly matched tasks jointly or training a highly correlated subordinate task in an adversarial manner will only lead to dissatisfactory performance of the major task.

Decision about which approach to adopt is easy to make when relationship between tasks are semantically unambiguous, but non-trivial when how positive or negative that one task contributes to another is not clear. In contrast to traditional prior knowledge based task matching measuring, we intend to find a mathematical representation, the matching confidence score, to measure the matching degree of two tasks and make sure it supports real-time calculation. With the matching confidence score calculated in every step of training stage, we are enabled to create a unified formulation of both joint optimization and deep domain adaptation-like learning strategy which will be referred to as major-subordinate-task learning, and beyond that, adaptively choose a learning strategy which better fit our context thus maximize content-relative information's contribution to orientation estimation.

3. OUR APPROACH

In this section we introduce our approach to predict image orientation in major-subordinate-task learning framework in the following sequence. We first detail our definition for matching confidence score of the major task and subordinate task. After that, we describe the formulation of proposed major-subordinate-task learning framework and introduce the matched gradients weight multiplier, a critical module that supports our learning strategy while running the standard stochastic gradient descent learning algorithm.

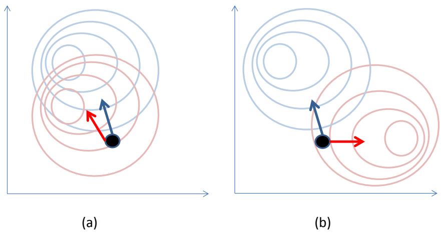

Fig. 1. Relationship between partial derivatives of the losses of different tasks with respect to parameter of feature extractor *i.e.* $\mathbf{w_f}$, when saddle points of different optimization tasks are (i)close to(a) or (ii)distant to(b) each other. Red and blue ellipses are contour of loss values of two tasks. Solid black oval denotes position of parameter $\mathbf{w_f}$ in parameter space $\mathcal{W}$. Red and blue arrows indicate directions of backward gradients from different tasks. We can see from the illustration that directions of gradients are similar when saddle points of different optimization tasks are densely located, otherwise the gradients will be dissimilar in direction.

3.1. Definition for Task matching degree

Generally, a deep learning model comprises two functional sections – a feature extractor composed of multiple lower layers and a task oriented network compounded with higher layers. In a convolutional architecture, deep features are extracted by a series of convolutional layers then fed into subsequent fully-connected layers for diversified machine learning tasks. By the definition of convolution operation, the features and their combinations are essentially responses of input to patterns parameterized with convolution kernels. Desirable feature extractor will be used to refer to a set of kernels over which the activations can optimally facilitate the ensuing task. Typically, desirable feature extractor is obtained after recursively updating the kernel parameters with stochastic gradients. Tasks with shared input are well matched when desirable feature extractors for these tasks are similar, no matter how different the way the extracted features are further utilized. Apparently, such measurement of matching degree of tasks is not calculable without an entire training phase.

Though similarity between desirable feature extractors is implausible to be attained step-wise, it is somewhat indicatable by the consistency of directions of updates which are applied by different tasks to current extractor, when features extracted by the latter is shared by multiple tasks. Let consider $\mathbf{w_f}$ the parameter vector of shared feature extractor, $\mathbf{w_{f1}}$ the desirable feature extractor for task one and $\mathbf{w_{f2}}$ the desirable feature extractor for task two and treat them as three points in

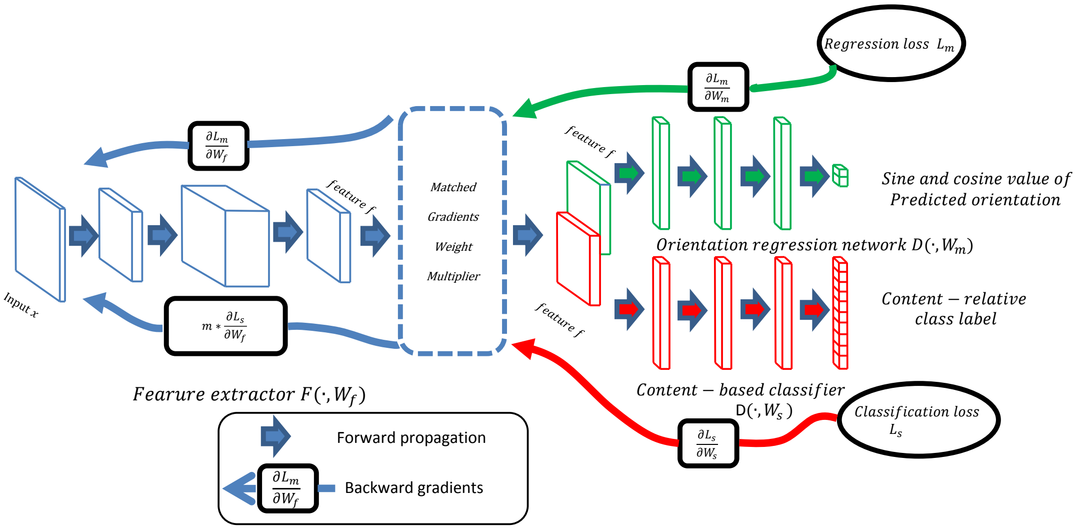
Regression loss L_m
$\frac{\partial L_m}{\partial W_m}$
$\frac{\partial L_m}{\partial W_f}$
feature f
Matched
Gradients
Weight
Multiplier
feature f
Sine and cosine value of
Predicted orientation
Orientation regression network $D(\cdot, W_m)$
Content − relative
class label
Input x
$m * \frac{\partial L_s}{\partial W_f}$
feature f
Fearure extractor $F(\cdot, W_f)$
Content − based classifier
$D(\cdot, W_s)$
Classification loss
L_s
$\frac{\partial L_s}{\partial W_s}$
Forward propagation
$\frac{\partial L_m}{\partial W_f}$
Backward gradients

Fig. 2. The proposed architecture for major-subordinate-task learning framework. In forward propagation, the input sample $\mathbf{x}$ is firstly mapped to feature $\mathbf{f}$ in feature space $\mathcal{F}$, which is then doubled and fed to two ensuing task-oriented networks, one copy each. In backward updating, **matched gradients weight multiplier** collects downstream gradients, according to which it calculates matching confidence score, *i.e.*, m. Finally, it multiplies backward gradients from subordinate task by m and ensures our learning framework adaptively choose the optimal learning strategy according to matching degree of tasks.

parameter space $\mathcal{W}$. If $\mathbf{w_{f1}}$ locates nearby to $\mathbf{w_{f2}}$, optimization directions from $\mathbf{w_f}$ to $\mathbf{w_{f1}}$ and $\mathbf{w_{f2}}$ are potentially similar. A heuristic interpretation of this is shown in Fig. 1. Additionally, updates to shared feature extractor are propagated back according to the chain rule, therefore their directions are varied with directions of partial derivatives of different losses with respect of the shared features.

The above observations lead us to define matching degree of two tasks in a step-wise point of view, *i.e.*, the inner production of the directions of partial derivatives of tasks' losses with respect of the shared features. We name it "matching confidence score" and formulate it as

$$m_{t_1,t_2} = \langle \frac{\partial \hat{L}_1}{\partial \mathbf{f}}, \frac{\partial \hat{L}_2}{\partial \mathbf{f}} \rangle, \tag{1}$$

where footnotes t_1 and t_2 represent task one and task two, respectively, L_1 and L_2 are losses of the two tasks, $\mathbf{f}$ is convolutional feature shared by both tasks, caret denotes the direction of the vector, angle brackets represent inner production of two vectors.

Formulation of matching confidence score makes sense in two aspects. To begin with, using gradients rather than features updated with these gradients to calculate matching degree bridges the gap between definition of task matching degree in feature extractor term and implementation requirement of real-time calculability. On top of that, by the nature of inner production, if the input vectors are normalized, the output will be close to one when the inputs are similar, close to minus one when widely dissimilar, and close to zero when orthogonal. The three relationships between inputs mentioned above are corresponding to three kinds of relationships between tasks: (i) highly correlated and mutually facilitative, (ii) reciprocally inhibitive and (iii) mutually independent. We will show in Section˜ 3.2 that this property enables proposed major-subordinate-task learning framework to adaptively switch between joint optimization and deep domain adaptation-like learning strategy. It also plays a crucial part in masking update from subordinate tasks which are irrelative to and not gainful to be trained in parallel with the major task.

3.2. Major-subordinate-task Learning Framework

In this subsection, we elaborate how matching confidence score is embedded into the proposed learning framework to achieve the desirable adaptiveness in shifting training strategies. By analysing optimization objects for convolutional feature extractor and two task-oriented fully-connected networks, respectively, we can derive unified formulations for joint optimization as well as deep domain adaptation-like learning strategy in terms of these functional sections when two tasks are trained in parallel. The objective functions for feature extraction, major task and subordinate task are as follows,

$$\begin{aligned} I_f(\mathbf{w_f}) = \sum_{i=1}^{N} & \ell_{ma}(D_{\mathbf{W_m}}(F(\mathbf{x}_i,\mathbf{w_f})), l_{ma}(\mathbf{x}_i)) \\ & + \lambda \ell_{sub}(D_{\mathbf{W_s}}(F(\mathbf{x}_i,\mathbf{w_f})), l_{sub}(\mathbf{x}_i)), \end{aligned} \tag{2}$$

$$I_m(\mathbf{w_m}) = \sum_{i=1}^{N} \ell_{ma}(D(F_{\mathbf{W_f}}(\mathbf{x}_i), \mathbf{w_m}), l_{ma}(\mathbf{x}_i)), \tag{3}$$

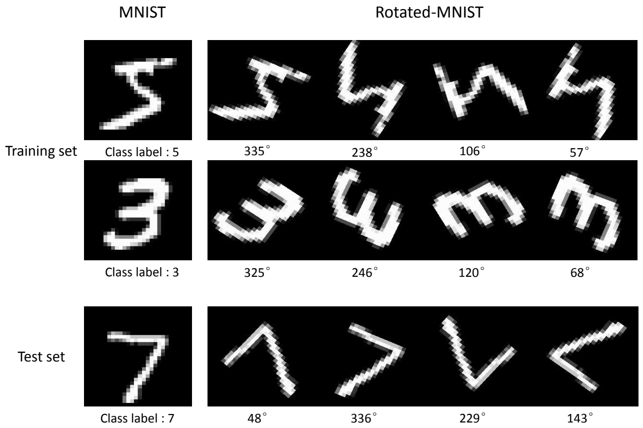

Fig. 3. Sample images. The left column is the original images in MNIST [18] and right columns are rotated images in **Rotated-MNIST**. We note at the bottom of every image the degree by which it is counterclockwise rotated. The first two rows are training examples while the last row is testing examples.

$$I_s(\mathbf{w_s}) = \sum_{i=1}^{N} \ell_{sub}(D(F_{\mathbf{W_f}}(\mathbf{x}_i), \mathbf{w_s}), l_{sub}(\mathbf{x}_i)), \qquad (4)$$

where $\mathbf{w_f}$, $\mathbf{w_m}$ and $\mathbf{w_f}$ denote network parameters for feature extractor, major-task-oriented network and subordinate-task-oriented network, respectively, F and D denote convolutional feature extractor and fully-connected network, parameters footnoted to network notation are fixed during optimization while updated if they appear in the brackets, $\mathbf{x}_i$ represents the i-th training example, $l_{ma}(\cdot)$ and $l_{sub}(\cdot)$ are mappings from samples to their corresponding labels with respect to major and subordinate task, ℓ_{ma} and ℓ_{sub} are individual loss functions for major and subordinate task.

The settings for two task-oriented networks are independent on training strategies, so the essential difference between joint optimization and deep domain adaptation-like learning strategy lies in the optimization objective for feature extractor, specifically, the parameter λ which controls trade off between the tasks. In joint optimization, λ is set positive so every branch in the architecture constructs a feed-forward structure. Contrarily, in the setting of deep domain adaptation-like learning strategy, a negative λ introduces adversarial training to one of the branches: stochastic gradients flowing downstream are reversed at the connection of feature extractor and task-oriented network, as a result, the extracted feature is made indistinguishable for the task [17]. On the other hand, we mentioned in Section 2 that joint optimization and adversarial training is desirable for mutually facilitative and reciprocally inhibitive tasks, respectively, and in Section˜ 3.1 we showed that matching confidence score serves as a perfect indictor for inter-task relationship. Combination of these ideas gives rise to our solution to the requirement of adaptiveness. We rewrite optimization objective for feature extractor in our

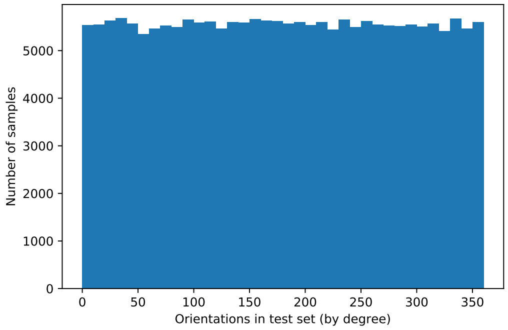

Fig. 4. The distribution of image orientations in our test set.

proposed learning framework as follows,

$$\begin{aligned} I_f(\mathbf{w_f}) = \sum_{i=1}^{N} & \ell_{ma}(D_{\mathbf{W_m}}(F(\mathbf{x}_i,\mathbf{w_f})), l_{ma}(\mathbf{x}_i)) \\ & + m_{t_m,t_s}\ell_{sub}(D_{\mathbf{W_s}}(F(\mathbf{x}_i,\mathbf{w_f})), l_{sub}(\mathbf{x}_i)), \end{aligned} \qquad (5)$$

Regarding orientation estimation the major task, content-relative classification the subordinate and running optimization (5), (3) and (4) seems has achieved our aim to switch training mode adaptively according to relationship between major and subordinate tasks: when close related, the matching confidence score is positive and tasks are jointly optimized; when mutually restrictive, the matching confidence score is negative and subordinate task is adversarially trained; when mutually independent, the matching confidence score is close to zero and the training mechanism equals to training orientation regression alone. However, it remains impossible to incorporate three optimization objects into a general feed-forward structure without corrupting independence between two task-oriented networks.

Thanks to the idea of gradient manipulation creatively brought up by Y. Ganin *et al.* [17], we are enabled to solve this problem with a module, **matched gradients weight multiplier**. It is inserted between feature extractor and two task-oriented networks and derives the architecture depicted in Fig. 2. Functions of the proposed module are (i) forwarding replicated features to task-oriented networks and (ii) multiplying updates of subordinate task to feature extractor by the matching confidence score calculated during training. Optimization (5), (3) and (4) can then be accomplished as running stochastic gradients descent for summation of losses of the major and subordinate tasks (*i.e.* $L_m + L_s$) on the proposed architecture. When training is finished, the feature extractor and orientation regression network will be connected directly to compose a desirable orientation estimator.

We would like to discuss two additional topics about the proposed major-subordinate-task learning framework. First, although matching confidence score is calculated with every training sample in every single step during the training stage, it should be statistically coherent if there is strong relationship

Table 1. Average prediction error of our method and counterpart settings

Method	Our Method	Joint	Adversarial	Naive
Average error(by degree)	2.72	2.97	2.91	2.91

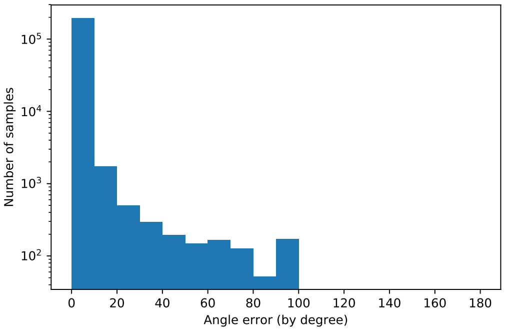

Fig. 5. The estimation error of our method.

underlying between the tasks. When tasks are loosely related, updates from major and subordinate task to the shared feature extractor should be statistically orthogonal. Therefore, training the model with mini-batch can smooth vibration which could be incurred by our learning strategy. Second, we believe it is reasonable that relationship between the major and subordinate tasks may vary in different phases of training, since nuance between desirable feature extractors for different tasks may lead to discrepancy between optimization directions at the rear of training stage. When a faithful subordinate task finally turns against the major, it is sensible to treat it reversely for the pursuit of optimized performance of the major task.

4. EXPERIMENTS

Introduction of our experiments is organized by three parts: we first introduce the settings for training & testing examples, then configurations for the architecture, finally the results against counterpart settings.

4.1. Evaluation Dataset

We evaluate the effectiveness of our method through a series of experiments. Since there are no datasets exclusively issued for image orientation prediction, we create **Rotated-MNIST** based on MNIST [18], a set of handwritten digits collected by Y. LeCun *et al.* by rotating every image example in MNIST by 20 random degrees. We treat orientations of original MNIST images as "appropriate" in our experiment. Consideration for the assignment of "appropriate orientation" is as below, to begin with, appropriate orientation for a kind of digits does exists inasmuch as orientations of original hand-written digits are largely determined by visual comfortableness which is mostly similar across writers. In this sense, aligning rotation-distorted images to what they used to be essentially maps them to the domain in which deep convolutional models are trained, as a result, our ultimate goal to improve performance of deep models will be achieved with aligned test samples. A packaged sine and cosine value with respect to the degree by which an image is rotated is used as its orientation label. Fig. 3 displays some sample images and their orientations defined in this way. In Fig. 4 we show that orientations of our created test set are evenly distributed.

4.2. Experiment setup

In our experiment, we base our proposed architecture on AlexNet [2]. Specifically, input images are reshaped to 227×227 and augmented to 3-channel. The feature extractor is composed of five convolutional and pooling layers with sizes $27 \times 27 \times 96$, $13 \times 13 \times 256$, $13 \times 13 \times 384$, $13 \times 13 \times 384$ and $6 \times 6 \times 256$. Orientation estimator comprises three fully-connected layers with sizes $1 \times 1 \times 4096$, $1 \times 1 \times 4096$ and $1 \times 1 \times 2$. The content-based classifier is identical to orientation estimator in terms of first two layers while shape of its third fully-connected layer is $1 \times 1 \times 10$. The output of orientation estimator is further unit-normalized to ensure the validity of sine and cosine values. We adopt l_2 distance as loss function for orientation estimation task. The output of content-based classifier is softmax regressed for cross-entropy loss function.

The algorithm we evaluated in experiments is same with what we have introduced in Section 3 except for the formulation of matching confidence score. Instead, we modify the original formulation by a linear transformation as follows,

$$m^*_{t_1,t_2} = \lambda_1 \langle \frac{\partial \hat{L}_1}{\partial \mathbf{f}}, \frac{\partial \hat{L}_2}{\partial \mathbf{f}} \rangle + \lambda_2. \tag{6}$$

From an experimental point of view, this modification makes it convenient to evaluate the performances of counterpart orientation estimators on our proposed architecture. With λ_1 and λ_2 set to zero, the proposed architecture is equivalent to a mono-branch AlexNet; setting $\lambda_1 = 0$ and $\lambda_2 = 1$ or $\lambda_2 = -1$, our architecture takes on joint optimization and deep domain adaptation-like learning strategy, respectively. While in the physical interpretation term, scaling controls trade off between the feedback from the major and subordinate task while shifting enables us to add prior knowledge of inter-task relationship to training strategy switching. For instance, we can assign λ_2 a positive value when two tasks are semantically related or assign λ_2 a negative value when tasks are semantically mutually exclusive. This is an interesting topic we would like to explore in the future. To make the comparisons more straightforward, we only use the setting $\lambda_1 = 1, \lambda_2 = 0$ to represent major-subordinate-task learning.

Other configurations for our experiments include: initial learning rate is set to 0.0001 and decay 4% every training step, size for minibatch is set to 60.

4.3. Results and discussions

We use the average error of orientation (by degree) as the criteria for evaluation. Table 1 reports the comparison of our method and counterpart settings. Joint optimization and deep domain adaptation-like learning strategies are written as "Joint" and "Adversarial" for short. We also use "Naive" to represent mono-branch orientation estimator based on AlexNet, which is identical to the proposed architecture used in testing stage. Apparently, introducing our methodology brings improvement of the estimation accuracy by decreasing prediction error by around 6.5%. We also show the overall prediction error of our method in Fig. 5. From the angle error histogram, we can see that estimation error is mostly below 10 degrees. Considering the fact that test set consists of image examples with orientations evenly distributed over the range $[0, 360)$, our proposed framework is indeed an accurate orientation estimator with enough robustness in face of a wide range of orientation variation.

5. CONCLUSION AND FUTURE WORK

We have presented our methodology of a novel learning framework, the major-subordinate-task learning framework and compared its performance on image orientation estimation with other training strategies. But there are works worth exploring afterwards. To start with, we can connect our network with a convolutional network and study by how much can we improve a CNN's ability to deal with rotation variation. Apart from that, we can extend our methodologies to other contexts other than image orientation estimation and test its performance on other tasks.

6. REFERENCES

[1] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *IEEE. J. PROC.*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[4] Ross Girshick, "Fast r-cnn," in *ICCV*, 2015.

[5] Shaoyan Sun, Wengang Zhou, Houqiang Li, and Qi Tian, "Scalable object retrieval with compact image representation from generic object regions," *TOMM*, 2016.

[6] Wengang Zhou, Houqiang Li, Jian Sun, and Qi Tian, "Collaborative index embedding for image retrieval," *TPAMI*, Feb. 2017.

[7] Jie Sun, Wengang Zhou, and Houqiang Li, "Orientation estimation network," in *ICIG*, 2017.

[8] David G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.

[9] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, "Orb: an efficient alternative to sift or surf," in *ICCV*, 2011.

[10] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *ICCV*, 2011.

[11] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst, "Freak: Fast retina keypoint," in *CVPR*, 2012.

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu, "Spatial transformer networks," in *NIPS*, 2015.

[13] Richard A. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *ICML*, 1993.

[14] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang, "Deep learning face representation by joint identification-verification," in *NIPS*, 2014.

[15] Emily M. Hand and Rama Chellappa, "Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification," in *AAAI*, 2017.

[16] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *ICML*, 2008.

[17] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015.

[18] THE MNIST DATABASE of handwritten digits, "http://yann.lecun.com/exdb/mnist/," .