Fact sheet: ECCV 2020 ChaLearn Looking at People 1st Fair Face Recognition Challenge

This is the fact sheet's template for the ECCV 2020 ChaLearn Fair Face Recognition Challenge [1]. Please fill out the following sections carefully in a scientific writing style. Then, send the compressed project (in `.zip` format), i.e., the generated PDF, `.tex`, `.bib` and any additional files to `juliojj@gmail.com`, and put in the Subject of the email "ECCVW 2020 FairFaceRec Challenge / Fact Sheets", following the schedule and instructions provided in the Challenge webpage [1] (post-challenge/fact sheets).

I. Team details

- Team leader name: Haoyu Qin
- Username on Codalab: CdtQin
- Team leader affiliation: Columbia University
- Team leader address: Apt 3R, 118, W 109th St, NYC, NY
- Team leader phone number: 646-219-4183
- Team leader email: hq2172@columbia.edu
- Name of other team members (and affiliation):
- Team website URL (if any):

II. Contribution details

A. Title of the contribution

Our main method lies in two parts, multi-branch training and similarity distribution manipulation. We conduct classification on different branches. Besides, we estimate the similarity distributions for these branches and add constrains on them. These two parts contribute to both the accuracy improvement and the alleviation of bias among different groups.

B. Introduction and Motivation

In order to improve the capability and fairness of face recognition model, we apply a multi-branch training strategy that each branch for a group. Such a multi-branch training strategy allow the model to get access to a balanced data input. Furthermore, since the threshold in face verification depends on those hard samples, namely low similarity pair from one subject and high similarity pair from different subjects, we conducted an off-line hard pair sample mining with our pre-trained model. We draw positive and negative similarity distributions of these pairs and add constrains to them. These constrains will force the same kind of distributions among different groups be closer and the distance between positive and negative distributions be larger.

C. Representative image / workflow diagram of the method

Our pipeline that related to the similarity distribution manipulation is illustrated in Fig. 1.

D. Detailed method description

We utilize a modified ResNet-101 [2] as our backbone model. We double the number of channel for each convolution layer in blocks and introduced several SE-blocks [3] to enlarge and strengthen the model. Moreover, we introduce some branches, such as Indian branch and Southeast Asian branch, for underrepresented groups as a supplement. And for each branch, we use ArcFace Loss [4] to conduct the classification task.

We offline construct hard positive pairs and online select top-k hard negative pairs for each branch. We calculate the cosine similarity of these pairs and estimate the distribution in the same way as [5]. For the drawn distributions, we adopted three constrains, specifically kl_loss, order_loss and entropy_loss. Note that the former two losses are proposed in [5]. Here we write the formulation for these three losses in our method:

$$L_{kl} = \sum_{a,b,a\neq b} \sum_{i} p_i^a * \log(\frac{p_i^a}{p_i^b}) \tag{1}$$

$$L_o = \sum_{a} E(p_i^a) - E(p_i^a) \tag{2}$$

$$L_e = \sum_{a} \sum_{i} p_i^a * \log(p_i^a) \tag{3}$$

where p represents the distribution and a and b represents branches.

For L_{kl}, the loss measures the KL Divergence of those two similarity distribution (a and b in the formula). The lower L_{kl}, the more similar of those distributions. It is calculated for the same kind of distributions of two difference group. For example, we add such a constrain for the positive similarity distributions of dark female and bright female.

For L_o, the loss illustrates the difference of two distribution's expectation. Intuitively, we desire a large margin between positive and negative distribution. So, we apply this loss on the positive and negative similarity distribution for each branch.

For L_e, the loss calculates the negative entropy of a single distribution. The ideal face verification model gives out one when we input a positive pair and it gives out zero when we input a negative pair. Even though it is hard to achieve such an ideal model, it would be great if the similarity distribution is slim, which means the entropy of the distribution is high. Introducing such a constrain allow the similarity distribution near the threshold to have lower variance, which brings us with better separation.

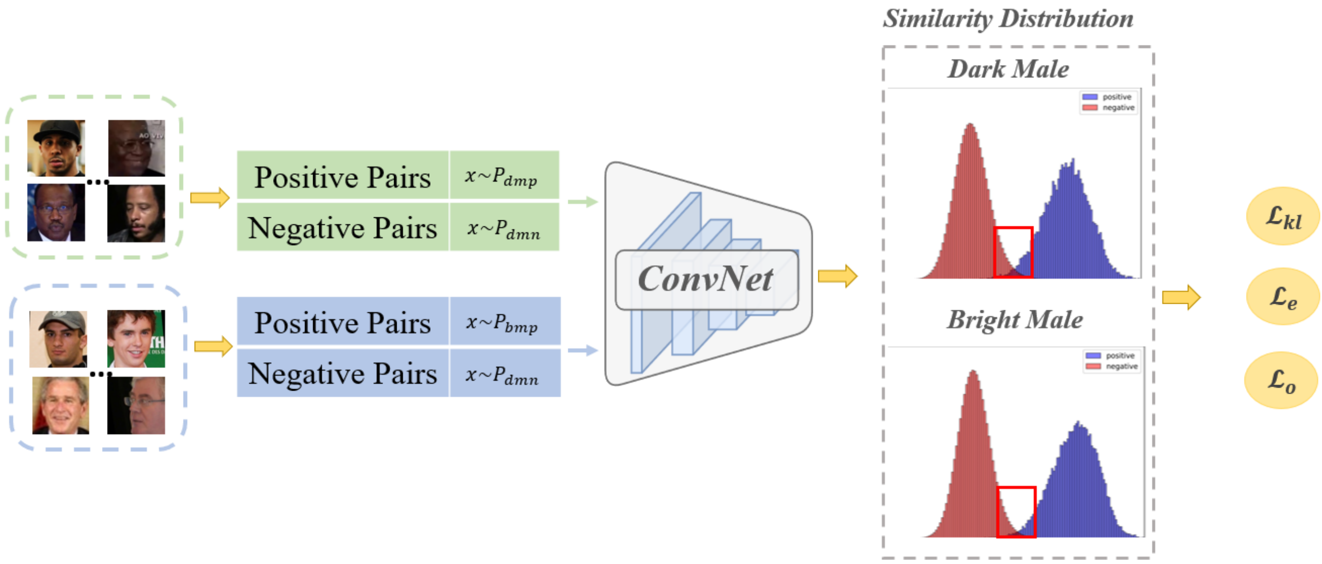

Fig. 1. Here we use dark male and bright male to demonstrate our pipeline. Three constrains, L_{kl}, L_e and L_o, are adopted to manipulate the similarity distribution for hard positive pairs and negative pairs. These constrains will separate the similarity distribution between negative and positive distributions and make the same kind of distributions among different groups to be closer.

Therefore, the total loss function for our method can be written as following:

$$L = L_A + L_{kl} + L_o + L_e \tag{4}$$

where L_A is the classification loss that uses ArcFace [4].

E. Challenge results and final remarks

Fill Table I with your obtained results, shown in the leaderboard of the challenge[1]. Note, if you joined the challenge in the test phase, keep the "development" row blank.

TABLE I

LEADERBOARD: RESULTS OBTAINED BY THE PROPOSED APPROACH.

Phase	Rank	Bias positive pairs	Bias negative pairs	Accuracy
Development	3	0.002334	0.000472	0.998477
Test	3	0.000405	0.000036	0.999827

III. ADDITIONAL METHOD DETAILS

Please reply if your challenge entry considered (or not) the following strategies and provide a brief explanation.

- **Did you use pre-trained models?** (✓) Yes, ( ) No
  If yes, please detail:
  A pre-trained face recognition model trained by arcface

- **Did you use external data?** (✓) Yes, ( ) No
  If yes, please detail:
  Several large private datasets that can provide a lot of subjects within underrepresented groups

- **Did you use other regularization strategies/terms?** (✓) Yes, ( ) No
  If yes, please detail:
  BN, and Weight Decay

- **Did you use handcrafted features?** ( ) Yes, (✓) No
  If yes, please detail:

- **Did you use any face detection, alignment or segmentation strategy?** (✓) Yes, ( ) No
  If yes, please detail:
  Private face detection and alignment. Several hard samples are hand cropped.

- **Did you use ensemble models?** ( ) Yes, (✓) No
  If yes, please detail:

- **Did you use different models for different protected groups?** ( ) Yes, (✓) No
  If yes, please detail:

- **Did you explicitly classify the legitimate attributes?** ( ) Yes, (✓) No
  If yes, please detail:

- **Did you explicitly classify other attributes (e.g. image quality)?** ( ) Yes, (✓) No
  If yes, please detail:

- **Did you use any pre-processing bias mitigation technique (e.g. rebalancing training data)?** (✓) Yes, ( ) No
  If yes, please detail:

[1] https://competitions.codalab.org/competitions/24123

Multi-branch training that each branch for one group and off-line hard sample mining.

- **Did you use any in-processing bias mitigation technique (e.g. bias aware loss function)?**
  (✓) Yes, ( ) No
  If yes, please detail:
  These branches are not only used for classification but also be used to estimate the distributions. We add constrains, namely kl_loss, entropy_loss and order_loss, on these estimated distributions. Here is a paper reference: `https://arxiv.org/abs/2002.03662`

- **Did you use any post-processing bias mitigation technique?** (✓) Yes, () No
  If yes, please detail:

  We extract features for both the original image and flipped image and calculate the mean of them as the final representation.

IV. Code Repository

Link to a code repository with complete and detailed instructions so that the results obtained on Codalab can be reproduced locally. This includes a list of requirements, pre-trained models, and so on. Note, training code with instructions is also required. This is recommended for all participants and mandatory for winners to claim their prize. **Organizers strongly encourage the use of docker to facilitate reproducibility**.

Code repository: `https://github.com/CdtQin/FairFace`

References

[1] ChaLearnLAP. ECCV 2020 ChaLearn Fair Face Recognition Challenge. [Online]. Available: http://chalearnlap.cvc.uab.es/challenge/38/description/

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[4] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[5] Y. Huang, P. Shen, Y. Tai, S. Li, X. Liu, J. Li, F. Huang, and R. Ji, "Distribution distillation loss: Generic approach for improving face recognition from hard samples," *arXiv preprint arXiv:2002.03662*, 2020.