Fact Sheet: ECCV 2020 ChaLearn Looking at People 1st Fair Face Recognition Challenge *Universidad Autonoma de Madrid (UAM): SensitiveLoss*

I. Team details

- Team leader name: Aythami Morales
- Username on Codalab: UAM_Ignacio
- Team leader affiliation: *Universidad Autonoma de Madrid (UAM)*
- Team leader address: Calle Francisco Tomás y Valiente, 11, 28049 Campus de Cantoblanco, Madrid, SPAIN
- Team leader phone number: +34914977558
- Team leader email: aythami.morales@uam.es
- Name of other team members (and affiliation): Ignacio Serna (UAM), Roberto Daza (UAM), and Julian Fierrez (UAM)
- Team website URL (if any): `http://biometrics.eps.uam.es/`

II. Contribution details

A. Learning Experience

We tested 4 competitive face detectors, 3 pre-trained models, and our new bias reduction method based on semi-hard triplet generation and selection: SensitiveLoss [1], [2]. The tested models are: VGG16 [3], ResNet-50 [4], and LResNet100E-IR [5]; trained with VGGFace2 [6] and MS1M-Arcface datasets [5]. In all 3 models we have slightly improved the accuracy (ca. 2%) and highly reduced the bias (between 31% and 68%) by incorporating our SensitiveLoss de-biasing technique.

B. Introduction and Motivation

Recently, as facial recognition systems have grown more sophisticated, their applications have expanded greatly. If we look at the most recent articles we see that the latest technologies in facial recognition seem to be touching the ceiling of perfect accuracy, almost no errors. The latest NIST FRVT report, as of May 22 of 2020, showed a False Non-Match Rate (FNMR) of 0.0301 @ False Match Rate (FMR) of 0.000001 on wild photos for face verification [7]. Based on this, it looks like the facial recognition problem is solved. In this context, bias and fairness across demographic groups is arising as an important research line in face recognition and biometrics at large [8].

Our goal in this Challenge on Fair Face Recognition is twofold: 1) to evaluate the performance of state-of-the-art face detection and face recognition algorithms in the framework proposed for the competition, and 2) to evaluate our recent de-biasing method called SensitiveLoss [1], [2]. The results demonstrate that the incorporation of our method further improves the excellent results of the state-of-the-art pre-trained models, both in terms of bias reduction and accuracy improvement.

C. Detailed Method Description

1) Face Detection: We have used 4 face detectors.

- MTCNN Detector [9]: this detector has a cascade structure with three stages of carefully designed deep convolutional networks that predict the face and five landmarks in a coarse-to-fine manner.
- dlib Detector: we use the dlib library, which has a pre-trained model based on a Convolutional Neural Network (CNN).
- Single Shot Detector (SSD): this is the detector available with the popular library OpenCV.
- RetinaFace Detector [10]: a robust single-stage face detector, which uses extra-supervised and self-supervised multi-task learning. RetinaFace simultaneously predicts the face score, the face bounding box, five facial landmarks, and the position and 3D correspondence of each facial pixel. It is trained in the Wider Face dataset [11], which consists of 32,203 images and 393,703 face bounding boxes.

Table I shows the detection errors for each detector and dataset. We can see very large differences between the OpenCV DNN, RetinaFace, and the other two algorithms. The results reported by the best approach (RetinaFace) show a detection error around 1.7%. These results suggest that face detection under challenging conditions is still an open problem. As we will show in the Section II-D, most of these errors are caused by large pose variations, occlusions, and low quality images [12]. Finally, we used the intersection of the 4 models, selecting the most centered face as the final one. The final percentage of images where a face was not detected by any of the methods was 0.54% (test set). In

TABLE I
Face Detection: Percentage of images in which no face is detected in the training, validation, and test sets.

Face Detector	Train	Validation	Test
MTCNN	21.59	22.55	22.39
Dlib CNN	21.70	21.58	21.35
OpenCV DNN	3.31	5.93	5.25
RetinaFace	1.72	1.72	1.62
Intersection 4 detectors	0.59	0.50	0.54

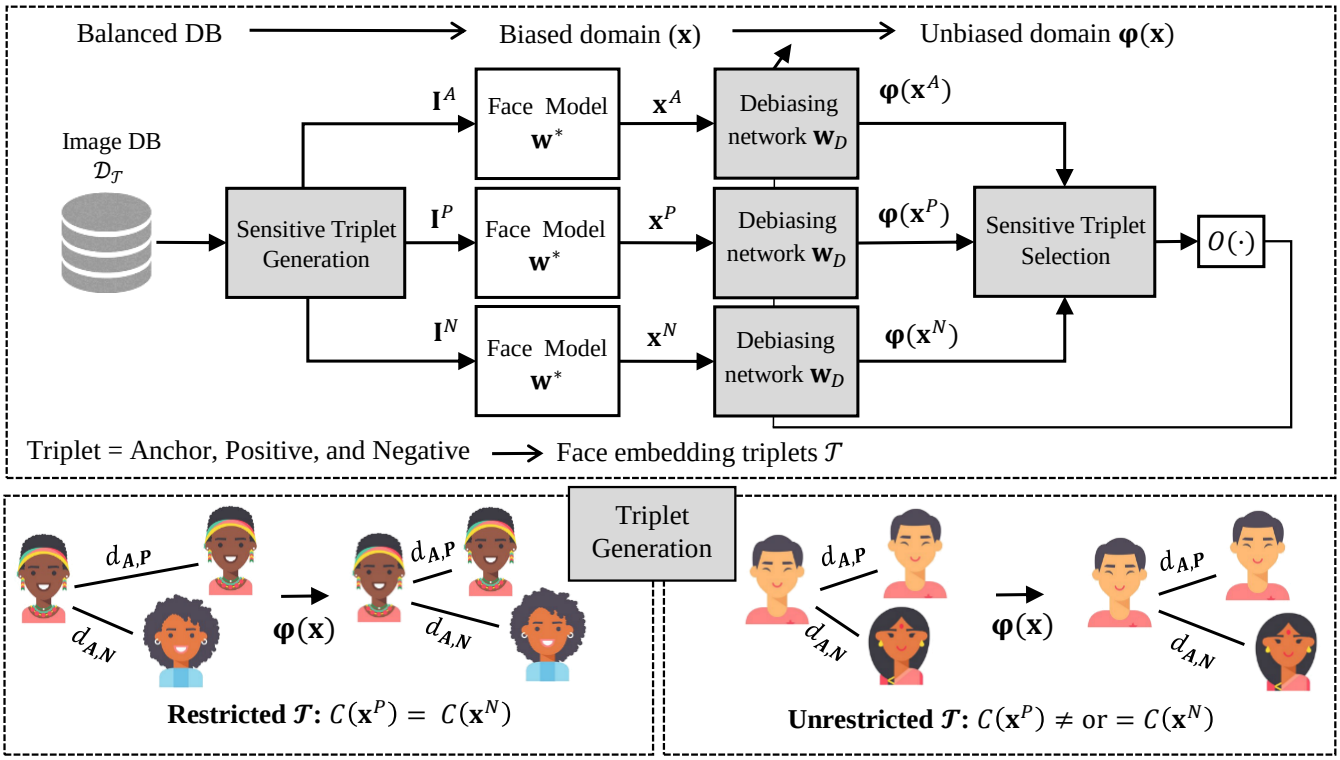

Fig. 1. (Up) Block diagram of the domain adaptation learning process that allows us to generate an unbiased representation $\boldsymbol{\varphi}(\mathbf{x})$ from a biased representation $\mathbf{x}$. A Balanced Dataset $\mathcal{D}_{\mathcal{T}}$ is preferable as input to train SensitiveLoss for selecting the triplets $\mathcal{T}$. This $\mathcal{D}_{\mathcal{T}}$ can be a different one or a subset of the (generally unbalanced) Dataset $\mathcal{D}$ used for training the biased model $\mathbf{w}^*$. (Down) Discrimination-aware generation of triplets given an underrepresented (unfavored) demographic group: the representation $\boldsymbol{\varphi}(\mathbf{x})$ increases the distance d between Anchor and Negative samples while reducing the distance between Anchor and Positive, trying in this way to improve the performance of the unfavored group.

order to provide a score for all images, when no face was detected, we subdivided the images using sliding windows with size equal to 30% of the full image, shifting them by 10%. The result was 49 subimages per image. When no face was detected, we compared with all the subimages and selected the best score.

2) Pre-trained Face Recognition Models: We have tested 3 competitive pre-trained models for facial recognition: VGG16 [3], ResNet-50 [4], and LResNet100E-IR [5].

- VGG16 [3]: this model comprises 5 convolutional blocks for a total number of parameters equal to 138M. This model was trained with the popular VGGFace2 dataset [6].
- ResNet-50 [4]: this model takes advantage of Residual Layers with 5 convolutional blocks and 5 identity blocks each containing 3 convolutional layers. The total number of parameters is 23M. This model was trained with the popular VGGFace2 dataset [6].
- LResNet100E-IR [5]: this is a heavy weight network with 5 convolutional blocks and 28 identity blocks each with 3 convolutional layers. The total number of parameters is 44M. This model was trained with MS1M-Arcface dataset and ArcFace loss [5]. This model was trained with carefully aligned faces. We have used the method InsightFace to align the images provided for the competition.

3) Bias Mitigation Learning: We have used the learning method proposed in [2]. It is a method focused on reducing the bias of highly competitive, pre-trained models. It works as an add-on to these models, and basically consists of adding a dense layer at the end of the pre-trained model. The dense layer has the following characteristics: number of units equal to the size of the pre-trained model output, dropout (of 0.5), linear activation, random initialization, and L_2 normalization. This layer is relatively easy to train (10 epochs and Adam optimizer) and is used to generate the new representation $\boldsymbol{\varphi}(\mathbf{x})$ (see Fig. 1).

The SensitiveLoss training method is based on a triplet loss function and online selection of sensitive triplets.

Assume that an image $\mathbf{I}$ is represented by an embedding descriptor $\mathbf{x}$ obtained by a pre-trained model. That image corresponds to the demographic class $C(\mathbf{x})$. A triplet is composed of three different images of two different people: Anchor (A) and Positive (P) are different images of the same person, and Negative (N) is an image of a different person. Anchor and Positive share the same demographic label, i.e. $C(\mathbf{x}^A) = C(\mathbf{x}^P)$, but this label may differ for the Negative sample $C(\mathbf{x}^N)$ (e.g. $C(\mathbf{x}^A) = C(\mathbf{x}^P) = \textit{Asian Female} \neq C(\mathbf{x}^N) = \textit{Caucasian Male}$). The transformation $\boldsymbol{\varphi}(\mathbf{x})$ represented by parameters $\mathbf{w}_D$ (D for De-biasing) is trained to minimize the loss function:

$$\min_{\mathbf{w}_D} \sum_{\mathbf{x}\in\mathcal{T}} (||\boldsymbol{\varphi}(\mathbf{x}^A) - \boldsymbol{\varphi}(\mathbf{x}^N)||^2 - ||\boldsymbol{\varphi}(\mathbf{x}^A) - \boldsymbol{\varphi}(\mathbf{x}^P)||^2 + \Delta) \tag{1}$$

where $||\cdot||$ is the Euclidean Distance, Δ is a margin between genuine and impostor distances, and $\mathcal{T}$ is a set of triplets generated by an online sensitive triplet generator that guides the learning process (see below for details).

Inspired in the semi-hard selection proposed in [3], [6], we propose an online selection of triplets that prioritizes the triplets from demographic groups with lower performances (see Fig. 1). On the one hand, triplets within the same demographic group improve the ability to discriminate between samples with similar anthropometric characteristics (e.g. reducing the false acceptance rate in *Asian Females*). On the other hand, heterogeneous triplets (i.e. triplets involving different demographic groups) improve the generalization capacity of the model (i.e. the overall accuracy).

During the training process we distinguish between generation and selection of triplets:

- *Triplet Generation*: this is where the triplets are formed and joined to compose a training batch. In our experiments, each batch is generated randomly with images from 300 different identities equally distributed among the different demographic groups (900 images in total). We propose two types of triplets generation (see Fig. 1):
  - **Unrestricted (U)**: the generator allows triplets with mixed demographic groups (i.e. $C(\mathbf{x}^A) = C(\mathbf{x}^N)$ or $C(\mathbf{x}^A) \neq C(\mathbf{x}^N)$). Thus, with 300 identities, around 135K triplets are generated (from which the semi-hard ones will be selected).
  - **Restricted (R)**: the generator does not allow triplets with mixed demographic groups (i.e. $C(\mathbf{x}^P) = C(\mathbf{x}^N)$). Thus, with 300 identities, more than 22K triplets are generated (from which the semi-hard ones will be selected).
- *Triplet Selection*: Triplet selection is done online during the training process for efficiency. Among all the triplets in the generated batches, the online selection chooses those for which: $||\mathbf{x}^A - \mathbf{x}^N||^2 - ||\mathbf{x}^A - \mathbf{x}^P||^2 < \Delta$ (i.e. genuine higher than impostor distance $\rightarrow$ difficult triplet). If a demographic group is not well modeled by the network (both in terms of genuine or impostor comparisons), more triplets from this group are likely to be included in the online selection. This selection is purely guided by performance over each demographic group and could change for each batch depending on model deficiencies.

D. Challenge Results and Final Remarks

See Tables II and III for our challenge results. It is important to note that we have limited the databases employed for training the pre-trained models and the debiasing method. We have intentionally not used the IJB-C database and others to avoid overlapping identities between the training data and the development and test data. Also, during the training process, we have not used any data other than the one provided by the Challenge.

TABLE II

LEADERBOARD: RESULTS OBTAINED BY THE PROPOSED METHOD.

Phase	Rank	Bias positive pairs	Bias negative pairs	Accuracy
Development	18.333	0.005009	0.010054	0.981019
Test	23.667	0.003478	0.008249	0.974710

Fig. 2. Examples of Test set images where any of the four detectors have detected a face. In the right lower corner we can see an error in the dataset.

In the development phase we have tested VGG16 and ResNet-50 models, improving slightly the accuracy and reducing significantly the bias by incorporating our SensitiveLoss de-biasing method. ResNet clearly outperforms VGG16 results. In the test phase we have added LResNet100E-IR and compared with ResNet-50, also managing to reduce bias and increase accuracy, and with very similar performances between both models, being slightly better LResNet100E-IR.

Figure 2 shows some of the face detection errors obtained for the Test set.

Figure 3 shows some of the genuine pairs with the lowest scores obtained by the ResNet-50 with SensitiveLoss approach and the Development set. We can see that lowest scores are caused by very low quality images, aging, pose, and occlusions. Note that errors in the database cannot be discarded.

We finally present Figure 4 to better understand the effect of our debiasing technique. The figure shows how the bias in the probability distribution of the impostors scores is drastically reduced.

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. LResNet100E-IR trained on MS1M-Arcface dataset[1].

[1]https://github.com/deepinsight/insightface

TABLE III

SCORES: RESULTS OBTAINED WITH AND WITHOUT OUR SENSITIVELOSS DE-BIASING METHOD FOR THE DIFFERENT TESTED MODELS.

Phase	Model	SensitiveLoss [2]	Bias positive pairs	Bias negative pairs	Accuracy
Development	VGG16	No	0.0409	0.0355	0.9334
Development	VGG16	Yes	0.0092 (↓78%)	0.0103 (↓71%)	0.9470 (↑1.5%)
Development	ResNet-50	No	0.0199	0.0202	0.9607
Development	ResNet-50	Yes	0.0050 (↓75%)	0.0101 (↓50%)	0.9810 (↑2.1%)
Test	ResNet-50	No	0.0213	0.0285	0.9504
Test	ResNet-50	Yes	0.0068 (↓68%)	0.0125 (↓56%)	0.9727 (↑2.3%)
Test	LResNet100	No	0.0052	0.0118	0.9751
Test	LResNet100	Yes	0.0035 (↓33%)	0.0082 (↓31%)	0.9747 (∼0.0%)

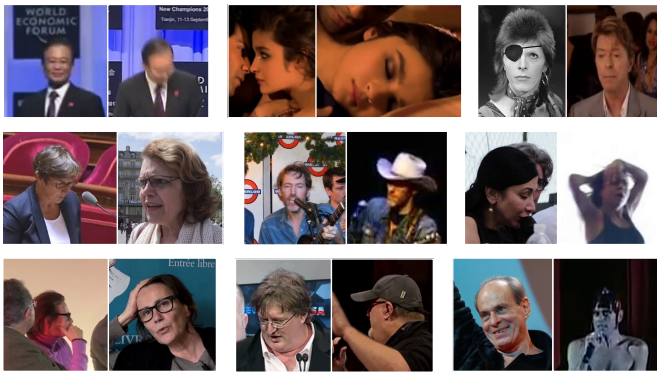

Fig. 3. Examples of genuine image pairs with the lowest scores of the Validation set.

ResNet-50 and VGG16 on VGGFace2 dataset[2].

- **Did you use external data?** No
- **Did you use other regularization strategies/terms?** No
- **Did you use handcrafted features?** No
- **Did you use any face detection, alignment or segmentation strategy?** Yes. For Face Detection we used all four:
  RetinaFace, Dlib CNN, OpenCV DNN, MTCNN. For the aligment: InsightFace.
- **Did you use ensemble models?** No
- **Did you use different models for different protected groups?** No
- **Did you explicitly classify the legitimate attributes?** Yes. We used the gender and skin color labels during training to create demographic batches to train the sensitive triplets.
- **Did you explicitly classify other attributes (e.g. image quality)?** No
- **Did you use any pre-processing bias mitigation technique (e.g. rebalancing training data)?** No
- **Did you use any in-processing bias mitigation technique (e.g. bias aware loss function)?** Yes. We used the SensitiveLoss function for discrimination-aware training [2].
- **Did you use any post-processing bias mitigation technique?** No

[2]https://github.com/rcmalli/keras-vggface

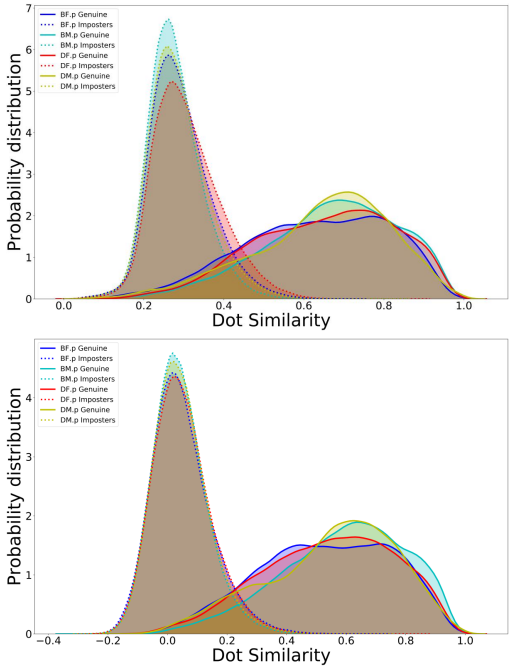

Fig. 4. Probability score distribution obtained for the best approach according to the different groups before (up) and after (down) application of the SensitiveLoss method. Impostor/Negative comparisons (dotted line), Genuine/Positive comparisons (continuous line) using the Development set.

IV. CODE REPOSITORY

Face detectors:

- RetinaFace: `https://github.com/deepinsight/insightface/tree/master/RetinaFace`.
- Dlib CNN: `http://dlib.net/cnn_face_detector.py.html`.
- OpenCV DNN: `https://github.com/opencv/opencv/tree/master/samples/dnn/face\_detector`.
- MTCNN: https: `https://github.com/ipazc/mtcnn`.

Face alignment: `https://github.com/deepinsight/insightface`.

Face Recognition models:

- ResNet-50 and VGG16 trained on VGGFace2 dataset: `https://github.com/rcmalli/keras-vggface`.

- LResNet100E-IR trained on MS1M-Arcface dataset: `https://github.com/deepinsight/insightface`.

REFERENCES

[1] I. Serna, A. Morales, J. Fierrez, M. Cebrian, N. Obradovich, and I. Rahwan, "Algorithmic discrimination: Formulation and exploration in deep learning-based face biometrics," in *AAAI Workshop on Artificial Intelligence Safety (SafeAI)*, February 2020.

[2] I. Serna, A. Morales, J. Fierrez, M. Cebrian, N. Obradovich, and I. Rahwan, "SensitiveLoss: Improving Accuracy and Fairness of Face Representations with Discrimination-Aware Deep Learning."

[3] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep Face Recognition," in *British Machine Vision Conference (BMVC)*, Swansea, UK, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016, pp. 770–778.

[5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive Angular Margin Loss for Deep Face Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A Dataset for Recognising Faces Across Pose and Age," in *International Conference on Automatic Face & Gesture Recognition (FG)*. Lille, France: IEEE, 2018, pp. 67–74.

[7] P. Grother, M. Ngan, and K. Hanaoka, "Ongoing Face Recognition Vendor Test (FRVT) Part 1: Verification," *NIST Interagency Report*, 2020.

[8] P. Drozdowski, C. Rathgeb, A. Dantcheva, N. Damer, and C. Busch, "Demographic bias in biometrics: A survey on an emerging challenge," *IEEE Transactions on Technology and Society*, vol. 1, no. 2, pp. 89–103, 2020.

[9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[10] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild," in *Conference on Computer Vision and Pattern Recognition (CVPR*. Seattle, Washington, USA: IEEE, 2020, pp. 5203–5212.

[11] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider Face: A face Detection Benchmark," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016, pp. 5525–5533.

[12] J. Hernandez-Ortega, J. Galbally, J. Fierrez, R. Haraksim, and L. Beslay, "Faceqnet: Quality assessment for face recognition based on deep learning," in *Proc. IAPR Intl. Conf. on Biometrics, ICB*, June 2019.