Fairness and Bias

Javier Montalvo Rodrigo
Roberto Alcover Couso

I. INTRODUCTION

This is our report for the labs based on detecting the effects of biases when training neural networks. The report will follow the structure of the labs.

Starting with a brief introduction of the dataset which we are going to use for the project and a few introduction to how the dataset is going to fit into each experiment. Following with the experiments in the work section, which will dive into each experiment the results and insight which can be extrapolated. The experiments will be presented as follows:

- TSNE of gender detection on different ethnicity groups: The goal is to see if a model trained only to disseminate gender internally is using ethnicity information.
- Biases in training and evaluation: The goal in this experiment is to analyze the performance of a model in demographic groups rather than giving just an accuracy or confusion matrix without fine grained detail, therefore retrieving more information of where the model may be suffering from biases.
- Activation functions: Analyzing the clustering capabilities of triplet loss and softmax, studying the distribution over the embedded space. Furthermore instead of using TSNE which is a unsupervised algorithm

Finally we will conclude with a brief conclusion section which will aim at summarizing the insights of the report.

II. DATASET

In this section we will introduce the dataset and how it's structure and labels are going to fit into each experiment of the report.

For this work we've been provided with the DiveFace dataset which contains faces from 6 demographic groups (3 ethnicities and two genders per ethnicity), Figure 1 shows some examples of each demographic classes defined, in order of appearance (black male, black female, asian male, asian female, white male and white female).

In this project we will make usage of the different subgrups of the dataset in order to:

- Understand latent information, such as ethnicity information when using a gender recognition system, where we will use TSNE to embed into a 2D space, this dimension is chosen so we can visualize the results, the embedding from a face recognition neural network in order to understand if even without the ethnicity information, the network is clustering the database by ethnicity.

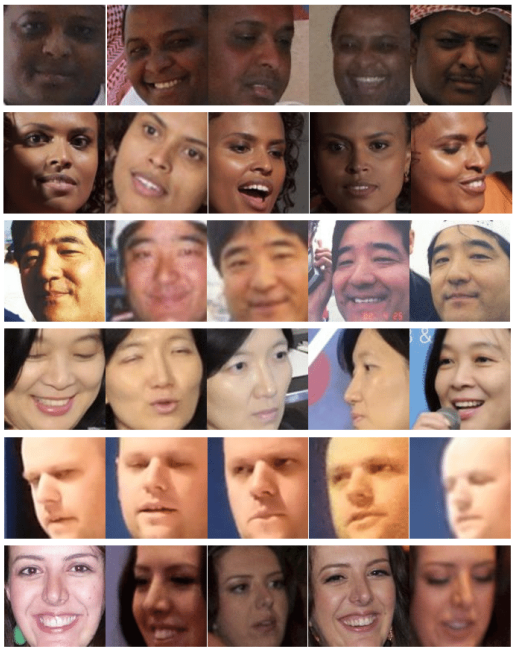

Fig. 1. DiveFace dataset, each row contains examples of each demographic group from [1]

- See biases in gender recognition, using the ethnicity labels we can measure the accuracy per ethnicity class, therefore understanding the actual performance of the network.
- Embedding space usage of different activation functions, using a demographic classifier and a 2D bottleneck we can plot the embedding of the bottleneck and understand how the network is distributing the space when using each activation function.

III. WORK

A. Face Recognition

For the first task, a network retrained for the task of face recognition was used to check how, even when the network was not trained for it, it could still create clusters for other classes, based on race and gender.

Using T-SNE algorithm, we can clearly see this separation happens in our face recognition embeddings:

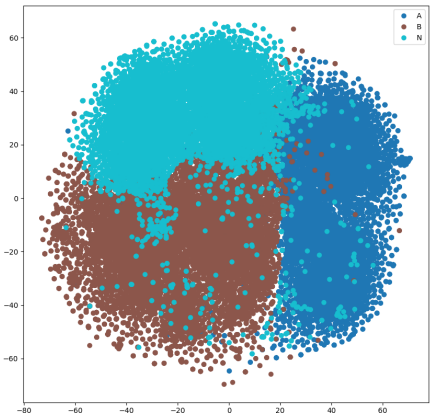

Fig. 2. TSNE over face recognition embeddings. Labeled by race.

And in fact, we can see this is true also with the gender of the people in the image:

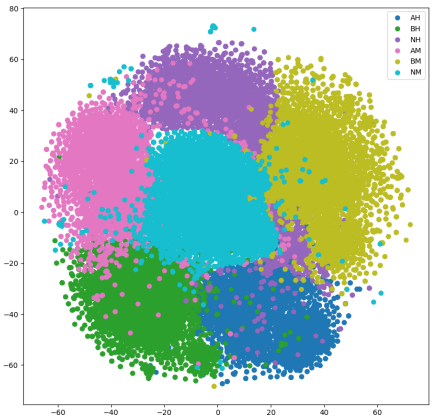

Fig. 3. TSNE over face recognition embeddings. Labels are demographic groups (gender and race).

Therefore, we could easily create classifiers for race or gender using these embeddings, as subjects from the same demographic group are close in this space.

B. Gender Recognition

For this task, three models were trained with the task of gender recognition, but only using subjects from different demographic groups for each training. One classifier was trained using asian people, another using black people and the last one using white people, and then, they were evaluated over the other demographics groups. Also, a classifier was trained with a balanced version of the dataset with training samples of each gender and race. All classifiers were trained for just three epochs, as the purpose of this exercise is noticing differences and not achieving the best accuracy. These results are collected in Table I.

	Accuracy (%)		
Classifier	**Asian**	**Black**	**White**
Trained with Asian	98.76	94.91	92.28
Trained with Black	96.57	97.80	95.35
Trained with White	70.04	84.61	95.73
Trained with Balanced Dataset	97.82	96.65	97.80

TABLE I

ACCURACY FOR DIFFERENT GENDER CLASSIFIERS.

As we can see, depending on the training dataset, results perform noticeably different when classifying genders with subjects from other races. This could be due to the network focusing on different features that may not be as general in all races, for example, black women often prefer to cut their hair, resembling those boys have, so a classifier for gender trained over images of black people, may not select *"hair length"* as a feature to distinguish between genders. Therefore, a classifier trained over white or asian people that may use hair length as a feature to classify with, will obtain poorer results when classifying images of black people.

In Figure 4 we include the activations maps of models trained over different subsets of the dataset (by race) when they are presented examples of different demographic groups. From these activations we can extract somee relevant information. For example, different classifiers have different activation intensities and regions over different subjects, with the most noticeable cases being the classifier trained over white people, that focuses on a side of the face to classify the black male example, and the balanced dataset that determines the white female is actually a female by focusing on her long blond hair and on her chin instead of in the center of the face like the other models do on the other examples.

In the case of the Asian female subject, every classifier shows a different activation region, which clearly shows the detected features vary noticeably between the different classifiers.

It is also interesting how the two ones with the most similar activation regions are the classifier trained over the black people subset and the classifier trained over a balanced subset, which are also the two models with the best inter class accuracy of the four.

In general, from those images we can see that biased classifiers have bigger activation regions over the subjects from the demographic were trained with, and the balance classifier has (except for the white female subject) a bigger average activation region than the biased ones.

C. Activation functions

For this task the goal is to analyze the differences in the embedded space when using a classical function like a Softmax and a metric learning function like the triplet loss.

Classical functions usually induce a usage of the embedded space due to the activation function, using as an example the Softmax activation function, which goes as follows:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

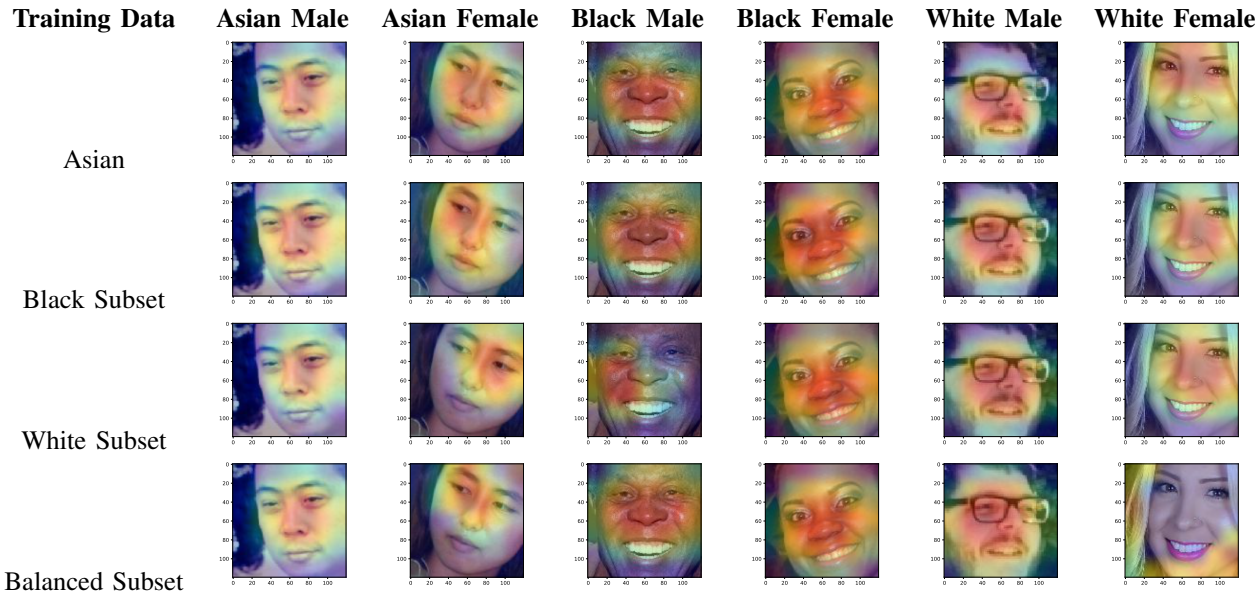

Fig. 4. Activations examples for different models over different demographic groups.

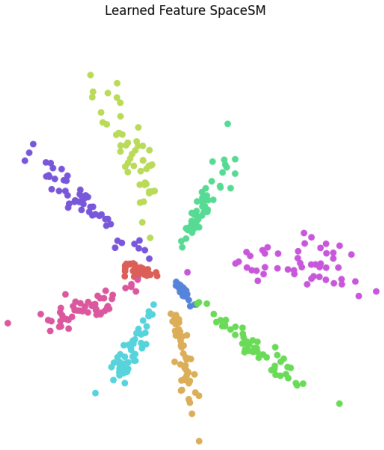

Fig. 5. 2D embedding from a overfitted problem with Softmax activation

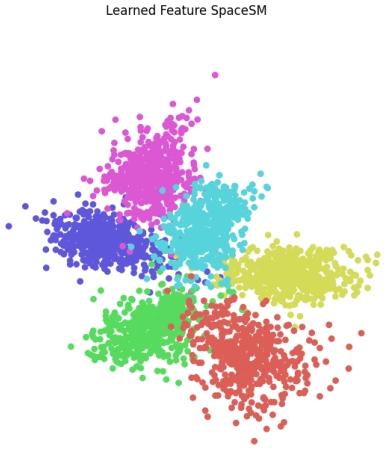

Fig. 6. Softmax embedding on the Gender Recognition dataset

This exponential induces in the Softmax activation function a radial dispersion of the embedding where each class will be allocated in a radius of the complex sphere induced by the exponentials (see figure 5), although elegant, simple and proven usefull, the softmax loss function does not explicitly optimise the feature embedding to enforce higher similarity for intra-class samples and diversity for inter-class samples. Therefore, in some scenarios like face recognition, gender identification, ethnicity and so, where large intra-class appearance variations due to pose variations and age gaps are inherent to the problem, the Softmax approach produces noticeable ambiguity in decision boundaries.

The ambiguity can be seen better when training with the dataset, where some examples are set to different clusters.

In contrast the triplet loss works by learning more robust

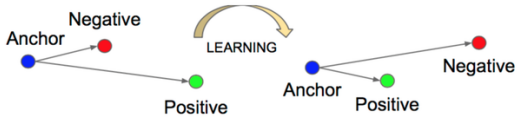

Fig. 7. Triplet loss representation

feature representation of the data, where examples of the same class are close together on the feature space and examples belonging to different classes are further apart.

The triplet loss works by comparing 3 images at a time: an anchor, a positive (image of the same class) and a negative (image of a different class). It aims at minimizing the distance between the anchor and the positive while maximizing the distance from the anchor to the negative (see

figure 7)

The triplet loss is written as:

$$\mathcal{L}(A, P, N) = \max\left(\| \mathrm{f}(A) - \mathrm{f}(P) \|^2 - \| \mathrm{f}(A) - \mathrm{f}(N) \|^2 + \alpha, 0\right)$$

where the goal is to maximize the euclidean distance between different classes and minimize the distance between examples of the same class. Table 8 illustrates the distribution of the embedding space as the training advances, starting from a random distribution of the examples, and as we follow the clusters of each class are defined and separated.

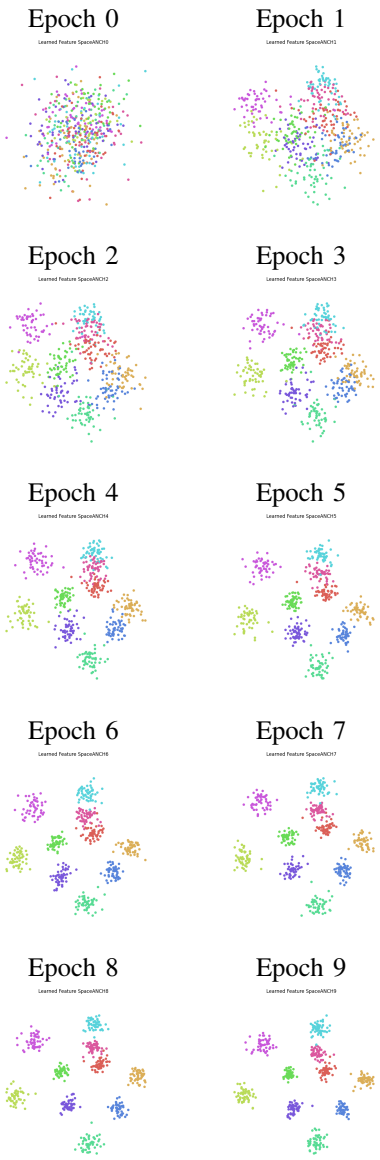

Fig. 8. Evolution through the training epoch of the embedding space when using a triplet loss

Due to the nature of working on the embedding space by separating classes the dispersion of the class is not imposed, the only imposition is the intra-class examples to be closer than any other example from other class.

Furthermore, due to the nature of the optimization, where in triplet loss works by directly optimizing the embedding space, improvements in the model such as hard negative mining, where hard triplets (Triplets that are not in the correct configuration, where the anchor-positive similarity is less than the anchor-negative similarity) could be used in the triplet to force the separation between those classes in said exemplars.

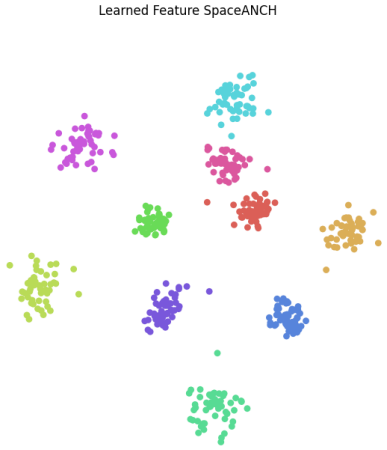

Fig. 9. Final embedding generated by the triplet loss

As we can see in Figure 9, the embedding has higher intra-class separation, compared to the softmax embedding from Figure 6.

IV. Conclusions

From this work we can extrapolate that when working with unstructured data such as images, sensitive information can be extrapolated by the network such as ethnicity, even without direct labels introduced into the training, therefore in depth analysis of whether or not a network is suffering of certain biases, furthermore in order to avoid sensitive information to be used in your model specific training mechanisms should be applied like debiasing, which includes biases into the training aiming at compensating or removing the usage of sensitive information of the model, one example being SensitiveNets [1] which uses adversarial learning to measure the impact or the amount of sensitive information, which are then introduced into the triplet loss as regularizing terms.

References

[1] Morales, Aythami Fierrez, Julian Vera-Rodriguez, Ruben. (2019). SensitiveNets: Learning Agnostic Representations with Application to Face Recognition.