Beyond Accuracy: Addressing Underestimation Bias in Multi-Label Image Classification through Multi-Objective Optimization

William Blanzeisky and Pádraig Cunningham

School of Computer Science
University College Dublin
Dublin 4, Ireland

Abstract. The adoption of Convolutional Neural Networks (CNNs) in various image classification tasks has led to a recognition of their inherent biases. This paper introduces a two-step approach to address underestimation bias in CNNs, specifically for multi-label image classification. Our focus is within the celebA dataset, known for gender fairness issues. We begin by defining fairness as an additional criterion in the CNN model training, adopting a hybrid-optimization strategy to generate a Pareto set of models, each demonstrating a different accuracy-fairness trade-off. Initially, we employ a gradient-based optimizer to train a robust CNN model solely on accuracy, establishing a strong base model for further refinement. Then, we use Multi-objective Particle Swarm Optimization (MOPSO) to fine-tune the weights of the fully connected (FC) layers in the CNN model, increasing emphasis on fairness while leveraging the base model's accuracy. To manage the complex nature of multi-label classification, we implement a dynamic weighting scheme to balance accuracy and fairness. But this presents a new challenge: improving fairness for one label can unintentionally make it worse for another. To tackle this, we propose a multi-task learning strategy, assigning each class label a dedicated FC layer, thus improving task-specific performance by reducing bias while maintaining adequate overall generalization accuracy.

1 Introduction

Although CNNs have shown notable success in various image classification tasks, they often suffer from inherent biases [16]. These biases emerge as unintended correlations with sensitive attributes, such as age, gender, and race [18]. This issue usually stems from CNNs being predominantly fine-tuned for generalization accuracy, inadvertently overlooking the possibility of discrepancies in the distribution of inaccuracies across sensitive groups. This discrepancy impacts model fairness, particularly in multi-label settings and brings into question the reliance on accuracy as the sole metric to be optimised. To address this, we propose a multi-objective approach to mitigate underestimation bias in multi-label image classification tasks.

Traditionally, CNN models are optimized for generalization accuracy. This optimization is typically done by using backpropagation in conjunction with gradient-based optimizers such as Adam [9]. Our method extends this optimization by fine-tuning the FC layers of the model using MOPSO, simultaneously optimizing for both accuracy and fairness. Our two-phase approach allows us to maintain the emphasis on accuracy while also incorporating fairness into the optimization process by effectively utilizing both gradient-based optimization and bio-inspired intelligence. Given the contradicting nature of these objectives and the complexity of multi-label image classification, we implement a dynamic weighting strategy to balance these objectives.

Nonetheless, our initial method exposed a recurring challenge, where enhancing fairness for one label inadvertently worsened it for another. To tackle this issue, we employ a multi-task learning approach, essentially creating a distinct classifier for each class label via task-specific FC layers while still ensuring parameter sharing in the hidden layers. With this approach, we significantly enhanced model performance, reducing underestimation bias without substantially compromising accuracy.

Following this introduction, we provide the background of the issue in section 2, providing an overview of three key areas: fairness in image labeling, multi-task learning, and multi-objective optimization. We begin our exploration by evaluating CNN models on the celebA dataset to highlight the issue of underestimation bias in section 3. Moving on to section 4, we introduce our proposed solution to the underestimation issue. To test how well our method performs within multi-label classification settings, we evaluate two strategies: dynamic weighting and a multi-task approach. We first look at the drawbacks of dynamic weighting in section 5. After that, we suggest a potential solution to this drawback using a multi-task approach in section 6. In the final part of the paper, section 7, we wrap up our discussion and suggest areas that could be explored in future research.

2 Background

Machine learning (ML) has become an integral part of many industries, leading to a growing need to understand and mitigate biases in ML models. With increasing legislation to prevent discrimination in AI systems, ensuring fairness in ML models is important. This section provides an overview of key concepts such as bias in image labeling, underestimation bias, multi-task learning, and multi-objective optimization.

2.1 Bias in Image Labeling

ML has greatly improved image labeling, boosting the accuracy of image classification based on content. Yet, this progress is not devoid of challenges, with biases in image classification emerging as one of the primary issues [18]. Recent studies divide the sources of these biases into two main categories: *data bias* and *model bias* [1].

Data bias comes from the training data used to build ML models. It's usually caused by factors like imbalanced sampling, which leads to overrepresentation or underrepresentation of certain groups, mislabeling that results in distorted image representation, and historical discriminatory practices that have unintentionally shaped the data.

Model bias, in contrast, happens when the learning algorithm either exacerbates inherent biases in the training data or creates new ones. This can be due to limitations in model capacity or uneven responses to the complexity of the learning problem, which can shift the model's focus towards certain data features [1].

2.2 Underestimation Bias

This paper focuses on model bias, especially when the model amplifies pre-existing data bias. This is often referred to as underestimation bias [7, 1]. It is a scenario where the model tends to underestimate the likelihood of less frequent outcomes, typically concerning minority groups, thereby exacerbating the bias already present in the data. We quantify the difference between what the model predicts and the real data distribution using an underestimation score, $\mathrm{US}_{S=s}$, which we calculate for a specific group S [1]:

$$\mathrm{US}_{S=s} \leftarrow \frac{P[\hat{y}=1|S=s]}{P[y=1|S=s]} \tag{1}$$

This ratio compares the favorable outcomes predicted by the classifier for the minority group to the outcomes actually existing in the data. If $S = 0$ signifies the minority group, a $\mathrm{US}_{S=0}$ score below 1 indicates that the classifier doesn't predict enough positive outcomes for the minority group.

An alternative underestimation score that considers divergences between the actual and predicted distributions for all groups S is the underestimation index (UEI) based on the Hellinger distance [7]:

$$\mathrm{UEI} = \sqrt{1 - \sum_{y,s \in D} \sqrt{P[\hat{Y}=y, S=s] \times P[Y=y, S=s]}} \tag{2}$$

Here y and s are the possible values of Y and S respectively. This Hellinger distance is preferred to KL-divergence because it is bounded in the range [0,1] and KL-divergence has the potential to be infinite. $\mathrm{UEI} = 0$ indicates that there is no difference between the probability distribution of the training samples and the prediction made by a classifier (no underestimation).

For instance, in a gender image classification scenario, an underestimation bias might occur if the model consistently underestimates the probability of identifying males in certain groups, such as those with long hair. In this case, UEI can quantify model bias by measuring the deviation of the model's prediction from the actual distribution across different groups.

Various strategies have been proposed for mitigating bias in ML in recent years [12]. These efforts target three stages of the model life cycle: pre-processing (transforming the dataset to eliminate biases), in-processing (modifying the algorithm's loss function to ensure fairness), and post-processing (adjusting model output to guarantee fairness).

Our method incorporates a Multi-Objective Optimization Problem (MOOP) strategy belonging in the in-processing category. A similar approach by Wang et al. combines a Relaxed Boundary Adaptation (RBA) strategy with a domain-independent approach [18]. The RBA strategy uses a process known as a Lagrangian relaxation iterative solver, which adds fairness constraints to the process, while the domain-independent approach tackles the Non-Discriminatory (ND) class-domain case specifically to reduce the correlation between class and domain. While our method aligns with the RBA strategy, we avoid using the Lagrangian method because it can create non-convexity when integrating fairness into the loss function. Instead, we use Multi-Objective Particle Swarm Optimization (MOPSO), which allows us to explore a range of models representing various trade-offs between accuracy and fairness, without needing to set a predetermined optimal fairness threshold.

2.3 Multi-Task Learning

Multi-Task Learning (MTL) is a learning approach that enhances model performance by leveraging shared information across interconnected tasks [19]. MTL mirrors human versatility in managing a variety of tasks simultaneously and has proven its effectiveness in a variety of fields, including natural language processing [2] and computer vision [14]. By forming shared representations among tasks, MTL improves both learning efficiency and prediction accuracy. In MTL, each class label has its own classifier, which allows the system to share common information across tasks while addressing each task's specific needs. This makes MTL useful for tackling complex challenges like ensuring accuracy and fairness in ML models for multi-label tasks.

The main idea of MTL is to reduce task-specific losses. Each task's loss is given a weight, which balances the losses among tasks [17]. These weights ensure that no single task's loss takes over the learning process. MTL allows multiple learning tasks to be optimized at once, bringing out both shared features and unique characteristics across the tasks.

An MTL model, represented as M, is defined by a set of parameters $\theta \in \Theta$. This set includes shared parameters θ_{shared}, which are the weights of layers shared across all tasks T, and task-specific parameters θ_t, which are weights for individual tasks. So, $\theta = \theta_{shared} \times \theta_1 \times \ldots \times \theta_T$.

Traditional MTL training aims to minimize multiple loss functions at the same time, with each function related to a different task:

$$\arg\min_{\theta}(L_1(\theta), \ldots, L_T(\theta)) \tag{3}$$

The main challenge in standard MTL training is finding the best model θ that minimizes all T tasks at the same time. This usually involves a scalarization approach, which combines all elements of the multi-tasking function into a single loss function. This method uses task-specific weights w_t, which show the relative importance of each task.

Our method takes a different approach from standard MTL training. We use MOPSO to individually fine-tune the task-specific layers while leaving the rest of the weights from the CNN model unchanged from the initial accuracy-only optimization process. This ensures our main model keeps shared weights across tasks while maintaining task-specific output layers. This separate optimization allows for parallel processing and removes the need to combine all losses to train the entire network at once, which is how traditional MTL works. More details about our approach are discussed in Section 6.

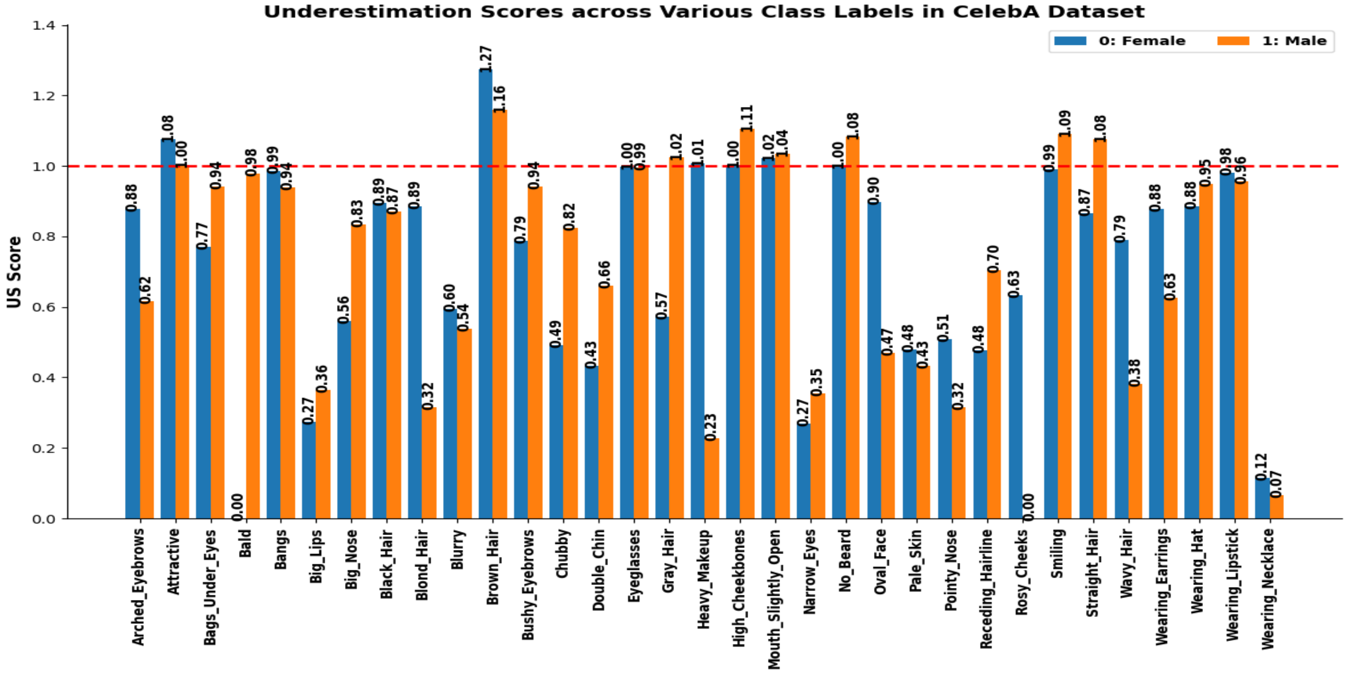

Figure 1: Underestimation scores for various class labels as predicted by a ResNet-50 model pretrained on ImageNet, evaluated on the (test) CelebA dataset. The figure showcases the correlation learned by the model between gender and certain attributes. Labels like 'Blond Hair', 'Oval Face', and 'Rosy Cheeks' are primarily associated with females, while 'Bald', 'Chubby', and 'Double Chin' are typically linked with males. Ideally, the underestimation score should approach 1 (indicated by the red line in the figure). The blue and orange bars represent the underestimation scores for females and males respectively. Scores below 1 indicate underestimation, while those above 1 suggest overestimation.

2.4 Multi-objective Optimization

Multi-objective optimization (MOO) focuses on the simultaneous optimization of multiple, occasionally conflicting, objectives. As real-world problems often involve multiple criteria, MOO becomes crucial for finding optimal solutions. In the context of our research, we use MOPSO to find a balance between accuracy and fairness. Balancing these objectives entails a trade-off, improving one of these objectives can sometimes reduce the other.

$$\min_{x}(f_1(x), f_2(x), ..., f_m(x)) \tag{4}$$

The equation above represents a MOOP, where multiple objective functions need to be optimized at the same time. Ensuring fairness while improving accuracy can be formulated as a MOOP. When there's no single solution that is the best for all criteria, Pareto optimality is used to find a set of non-dominated solutions [11]. These problems are complex, so approximation methods are often used to solve them. Several techniques, including meta-heuristics like MOPSO, have been proposed to tackle MOOPs [4, 6].

2.5 Multi-objective Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a stochastic optimization algorithm inspired by how birds flock [8]. It tries to optimize an objective function by first creating possible solutions, then iteratively directing each particle towards the best solution. The position and velocity of each particle are updated based on both individual and collective experiences:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{5}$$

$$V_i^{t+1} = wV_i^t + c_1r_1(pbest_i^t - X_i^t) + c_2r_2(gbest^t - X_i^t) \tag{6}$$

MOPSO is a variant of PSO that extends its applicability to MOOPs. MOPSO uses a global repository to record particles' experiences and Pareto dominance to determine flight direction [3]. How the repository is updated and how solutions are kept plays a big role in creating diverse Pareto fronts.

In the next sections, we detail our proposed method for reducing underestimation bias in multi-label image classification tasks, using the concepts explained in this background.

3 Exposing Underestimation Bias: A Deep Dive into the CelebA Dataset

The CelebA dataset, a large-scale face attributes database with more than 200,000 celebrity images, each annotated with 40 attribute labels, has become a critical benchmark for evaluating facial recognition algorithms [10]. However, this dataset is also known to exhibit inherent biases, a key concern for ML and more specifically for CNNs [18].

These biases lead to unequal performance between different demographic groups. Algorithms trained on the CelebA dataset often have variable prediction accuracy for different attributes or demographics [18]. For example, certain genders are under-represented, leading to lower predictive accuracy for these groups.

We demonstrate this bias by evaluating a ResNet-50 [5] model pretrained on the ImageNet dataset [15]. We evaluate its performance on 39 attributes from the Aligned & Cropped subset of CelebA, focusing on the "Male" attribute as the sensitive feature for fairness evaluation [10]. We considered 34 out of the 39 attributes in our analysis, based on having enough validation and test images.

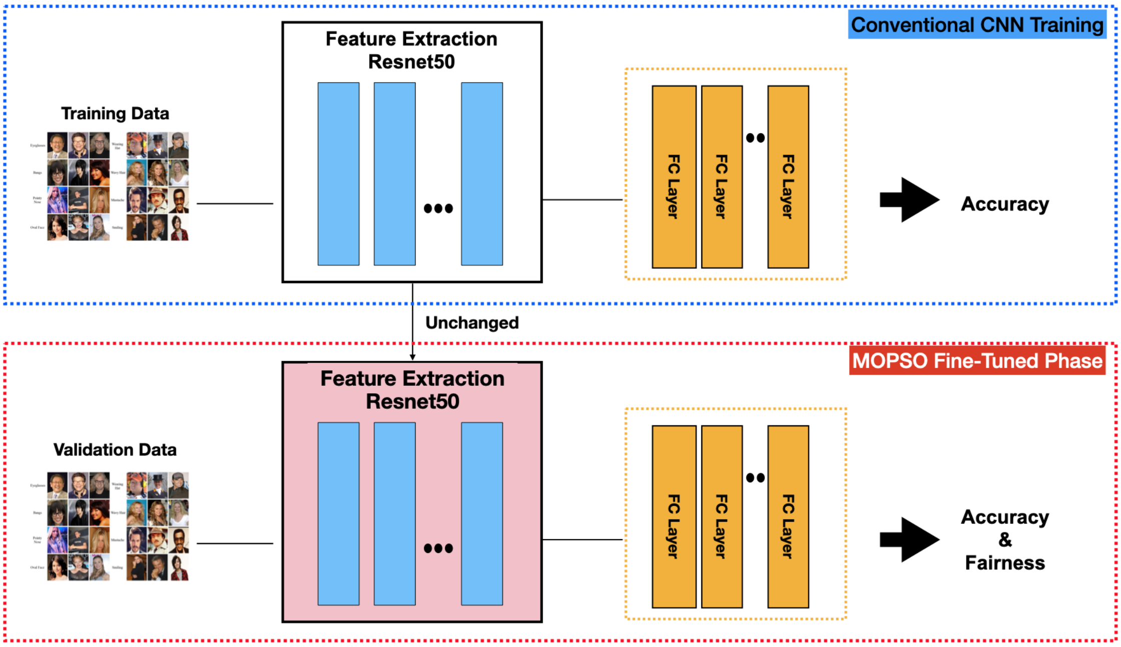

Figure 2: The framework of our two-stage approach for addressing underestimation bias in CNNs using a ResNet-50 base. The initial stage utilizes a gradient-based optimizer to train the CNN model on accuracy alone, establishing a robust base model. The subsequent stage applies MOPSO to the FC layers of the model, refining their weights using the validation set only, thus enhancing the model's fairness and leveraging the accuracy of the base model.

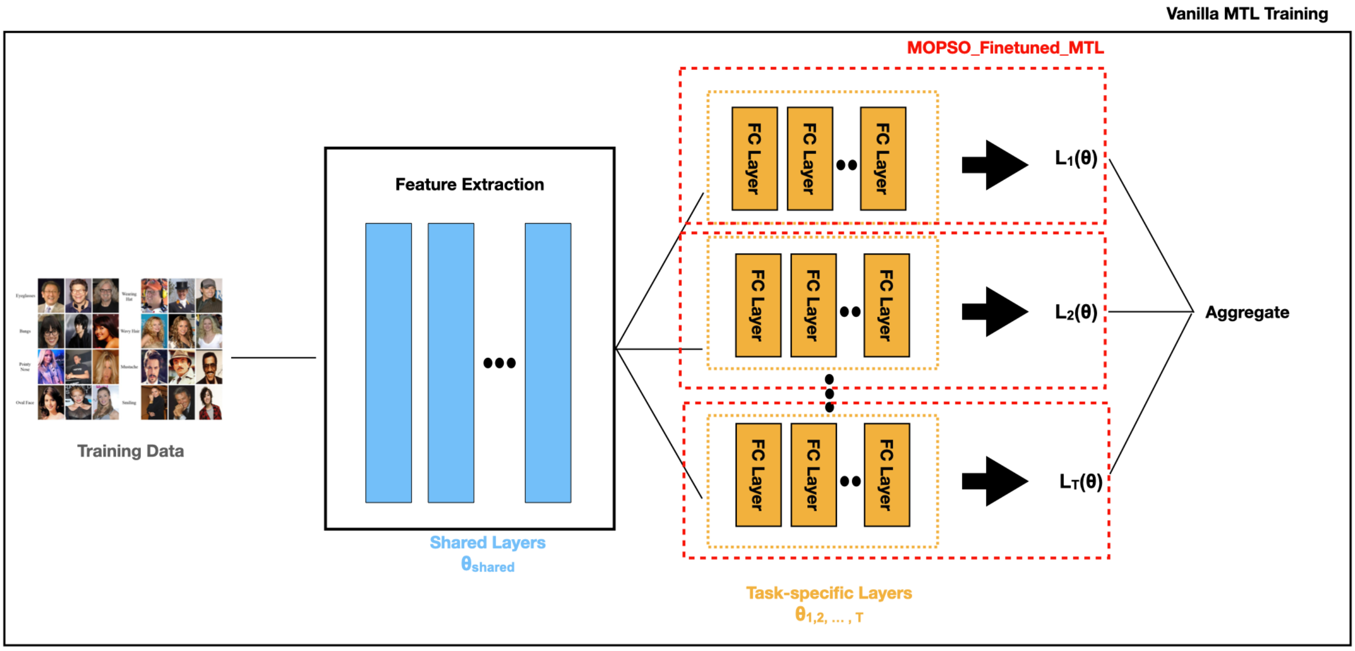

Figure 3: Comparison between MTL and MOPSO Method for Training MTL Models. Our multi-objective approach independently trains each task-specific layer in parallel using MOPSO, considering both accuracy and fairness objectives. This approach differs from traditional MTL, where the loss is aggregated for all task-specific layers. The figure highlights the shift from the traditional MTL approach to our MOPSO method, showcasing the parallel training of task-specific layers using multi-objective optimization for improved accuracy and fairness in multi-task learning scenarios.

We set our parameters following the methodology presented by [18], unless otherwise noted. In the ResNet-50 model, we replace the FC layers with two successive FC layers, each separated by a dropout layer and a Rectified Linear Unit (ReLU) activation function. For training, we use the binary cross-entropy loss with logits, with the Adam optimizer and a learning rate set at 10^{-4}. Our model is trained with a batch size of 32 for 50 epochs. We select the best model based on its performance over all epochs on the validation set.

Figure 1 shows the underestimation score. This score is the ratio between correctly predicted outcomes for the minority group and their actual representation in the dataset. Ideally, the underestimation score should be about 1, shown by the red line in the figure. The blue and orange bars show the underestimation scores for females and males. Scores below 1 show underestimation, while those above 1 show overestimation.

Figure 1 shows a clear link between certain labels and genders. Labels like "Blond Hair", "Oval Face", and "Rosy Cheeks" are mostly associated with females, while "Bald", "Chubby", and "Double Chin" are more linked with males. This discrepancy comes from how ML algorithms are usually trained: they often focus on overall accuracy without considering the distribution of remaining inaccuracies, especially when sensitive attributes like race or gender are involved.

Our underestimation concept aims to fix this problem, making these predictions more aligned with the actual distribution in the training data. This ensures more reliable prediction accuracy for underrepresented groups, like bald or chubby females, or males with oval faces or rosy cheeks. In the next section, we'll explain how our MOPSO strategy can help mitigate underestimation.

4 Using MOPSO for Mitigating Underestimation Bias

The central objective of our research, in addressing the issue of underestimation, is to align our predictions more precisely with the actual attribute distribution present in our training data. In this paper, we propose a strategy to mitigate underestimation bias by incorporating underestimation as an additional criterion during the CNN model training phase.

Including underestimation as an additional constraint within an algorithm's optimization function can result in a non-convex cost function. This situation presents considerable challenges for gradient-based optimizers like Stochastic Gradient Descent (SGD) or Adam, typically used in CNN training, because of the potential presence of multiple local minima or saddle points. To overcome these challenges, we propose a multi-objective optimization strategy that combines the benefits of both MOPSO and gradient-based optimizers like SGD or Adam.

4.1 Integrating MOPSO and gradient-based optimizer for CNN training

Bio-inspired evolutionary algorithms such as MOPSO offer strong exploration capabilities, but they can suffer from computational demands, hyperparameter sensitivity, and scalability issues in large search spaces. On the other hand, gradient-based optimizers like SGD and Adam are efficient, but they may struggle with multi-objective optimization due to the non-convex nature of the problem.

To leverage the strengths of both strategies, we introduce a hybrid approach. Initially, we use a gradient-based optimizer to train a robust CNN model, focusing only on accuracy to generate a strong base model - this will act as a starting point for MOPSO. Then, we use MOPSO to fine-tune the weights of the Fully Connected (FC) layers from the base model only on the validation set. This sequential approach benefits from the initial model's accuracy and ensures a more detailed exploration of the weight space with a focus on fairness. Figure 2 contrasts our innovative hybrid training strategy with traditional CNN training.

4.2 Overcoming Premature Convergence: A Modified MOPSO Approach

Our initial experiments with the traditional MOPSO algorithm revealed a tendency towards premature convergence, which can impede the search for global optimum solutions. To address this, we draw inspiration from the simulated annealing concept, introducing random Gaussian noise to the position of particles in densely populated areas in the objective space. This allows us to avoid local optima. If the disturbed particles produce a better solution, they become the new personal best. Conversely, if the updated solution is worse, it may still be accepted with a probability of $P_{accept} = 0.5$.

Parameter tuning for MOPSO is especially important due to the balance between exploration and exploitation. Key MOPSO parameters include the inertia weight w, and learning factors c_1 and c_2. Literature offers detailed analyses to find optimal parameters that enhance MOPSO's performance. In our research, the self-adaptive parameters strategy proposed by Montalvo et al. [13] proved to be effective. Therefore, we adopted it for setting our parameters. [1]

5 MOPSO with Dynamic Weighting

Given the multi-label nature of the CelebA dataset, we employed a dynamic weighting strategy to aggregate accuracy and fairness metrics across target labels. This strategy dynamically assigns weights that are inversely proportional to the accuracy and fairness associated with each respective attribute. The principal objective here is to direct the MOPSO optimization algorithm toward enhancing fairness across all attributes rather than concentrating solely on those with the highest UEI score.

Formally, given a dataset $\mathcal{D}(X, Y, S)$, wherein X denotes the feature vector, Y the target label, and S the sensitive attribute, we designate $\hat{Y}$ as the prediction output of a model $\mathcal{M}(\theta, X, S)$. The optimization problem can then be formally expressed as:

$$\theta = \arg\min_{\theta} \sum_{i=1}^{n} \left(w_{i,u} \cdot U(Y_i, \hat{Y}_i, S_i) \right), \left(-w_{i,a} \cdot A(Y_i, \hat{Y}_i) \right) \quad (7)$$

Where:

- n is the number of class labels
- $w_{i,U}$ and $w_{i,a}$ are weights for the ith class label for UEI and accuracy, respectively
- $U(Y_i, \hat{Y}_i, S_i)$ represents the UEI for the ith class label. We use UEI as the metric of underestimation to consider all possible combinations of class labels and sensitive attributes.
- $A(Y_i, \hat{Y}_i)$ represents the accuracy for the ith class label
- Y_i and $\hat{Y}_i$ are the actual and predicted labels for the ith class label, respectively
- S_i is the sensitive attribute for the ith class label

[1] The code implementation of our framework is available on our GitHub page https://github.com/williamblanzeisky/AddressingBiasinMultiLabelImageClassification.

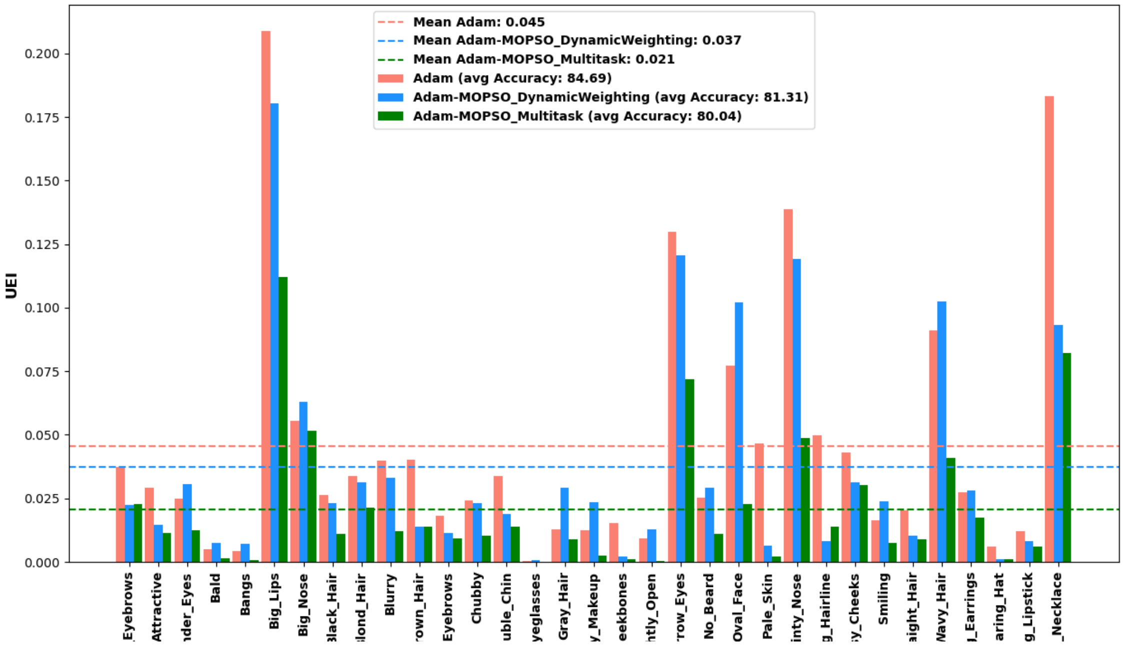

Figure 4: Evaluation of Different Approaches on Test Set. The red bar represents the results of a ResNet-50 CNN model trained solely on accuracy using Adam. It reveals the presence of underestimation bias, as indicated by high UEI scores across different labels. The blue bar represents MOPSO with a dynamic weighting strategy for optimizing both accuracy and fairness. Interestingly, while this approach partially mitigates underestimation, it unintentionally exacerbates performance issues in other attributes due to the use of weighted averages as objectives. Our MTL approach, represented by the green bar, effectively addresses the underestimation problem by reducing the UEI while managing a reasonable loss in accuracy (5%).

The weights $w_{i,U}$ and $w_{i,a}$ can be defined inversely proportional to the accuracy and fairness as:

$$w_{i,U} = \frac{1}{A(Y_i, \hat{Y}_i) + \epsilon}, \quad w_{i,a} = \frac{1}{U(Y_i, \hat{Y}_i, S_i) + \epsilon} \tag{8}$$

Here, ϵ is a small positive number added to prevent division by zero. The optimization problem thus involves finding the model parameters θ that minimize the loss function defined in Eqn. 7.

Although this method assists in reducing the dimension of the optimization search space and ensuring equal consideration of all attributes, our findings show that enhancing fairness or accuracy in one class label can inadvertently undermine performance on other attributes.

To illustrate this phenomenon, we conduct an experiment using a ResNet-50-based CNN, initially trained with the Adam optimizer to optimize for accuracy, using the entirety of the training data. We then transitioned to a fine-tuning phase, which exclusively refines the weights of the Fully Connected (FC) layers using MOPSO. This fine-tuning process optimizes both fairness and accuracy, with the weights derived from the initial accuracy-optimized training serving as the starting point for MOPSO's initial particle generation. Importantly, the fairness fine-tuning process utilizes only the validation set, which doesn't overlap with the training set, thereby enabling a reduction in training time.

To examine the impact of our framework in mitigating underestimation, we select the model with the lowest UEI from the Pareto set, tolerating minor losses in accuracy. Figure 4 illustrates this model's performance on a separate test set.

Interestingly, while our framework succeeds in somewhat fixing overall underestimation, it can unintentionally worsen the performance of the other attributes. This inadvertent consequence arises due to the use of weighted averages as the optimization objectives. This issue is more evident in attributes such as "Big Lips," where the initial UEI was significantly higher. This approach model excels at mitigating this, driving the UEI towards zero, but in doing so, it inadvertently escalates the UEIs for other attributes such as "Bald", "Oval Face", and "Gray Hair", all of which commenced with lower UEIs.

This problem gets worse and the number of attributes increases. With an expanding attribute set, the value tends to saturate when averages are used to aggregate the scores, or each attribute's contribution to the mean decreases. This situation underscores the necessity for a more sophisticated approach that accommodates the complex interplay of accuracy, fairness, and the diverse nature of attributes in multi-label datasets.

In response to this issue, we propose an alternative approach that employs multi-task learning. In the following section, we will demonstrate how this strategy can alleviate this problem.

6 Multi-task Learning: A Decoupled Approach

Understanding the multi-label character of the CelebA dataset, we propose a multi-task learning strategy that allocates a unique classifier for each attribute. In simpler terms, each task has its own separate section of the model that focuses on learning and predicting that specific category. By doing so, we can enhance the model's ability to accurately predict each label because it can learn from its specific task without being influenced by others. In essence, this method aims to separate the performance of individual attributes, promoting independent optimization.

We hypothesize that such separation could potentially fix the "whack-a-mole" problem encountered in the dynamic weighting approach. Unlike the previous approach detailed in equation 7, each task-specific layer optimizes both accuracy and UEI for its corresponding attribute. Thus, attribute-specific weights are independently calculated, reflecting the performance of their respective classifiers. This independent optimization helps strike a better balance between fairness and accuracy for each attribute and the model overall.

Typical multi-task learning involves combining losses from each output neuron, followed by backpropagation step, usually using gradient-based optimizers like Adam or SGD. This technique enables parameter sharing in hidden layers across all tasks, while dedicating the final FC layer for task-specific objectives, fostering specialization for each task. In alignment with our strategy outlined in Section 4.1, we take a step further by employing MOPSO to fine-tune the weights of task-specific layers. Our focus remains on optimizing for both accuracy and fairness. To preserve parameter sharing, we keep the weights of the initial CNN model, which was optimized for accuracy, unchanged. This unique optimization during the fine-tuning phase enables parallel processing and eliminates the need to combine all losses for entire network training at once. Moreover, the fine-tuning phase uses the validation set only.

This approach allows for the optimization of each attribute's accuracy and fairness independently, thereby providing a potentially more balanced solution to the underestimation issue. In fact, as shown in Figure 4, our multi-task learning strategy effectively mitigates underestimation while maintaining reasonable generalization accuracy (with a 5% drop), compared to the dynamic weighting method.

7 Conclusion

In this study, we present a multi-objective strategy to reduce underestimation bias in Convolutional Neural Networks (CNNs), with a focus on multi-label image classification tasks. We include underestimation as an additional objective in the CNN training process. To achieve an optimal balance between the accuracy and fairness of the multi-label task, we employ a dynamic weighting strategy. However, we encountered a problem—termed the "whack-a-mole" issue—where enhancing fairness for one label inadvertently compromises it for another. To solve this, we implemented an MTL approach, assigning a unique layer in the network to each class label. This allows us to fine-tune the performance of each label independently, mitigating the impact of the "whack-a-mole" problem. Our MTL approach demonstrated significant progress, substantially reducing underestimation bias while preserving adequate generalization accuracy.

Acknowledgements

This work was funded by Science Foundation Ireland through the SFI Centre for Research Training in Machine Learning (Grant No. 18/CRT/6183) with support from Microsoft Ireland.

References

[1] William Blanzeisky and Pádraig Cunningham, 'Algorithmic factors influencing bias in machine learning', in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, ed., Michael Kamp, pp. 559–574, Cham, (2021). Springer, Springer International Publishing.

[2] Shijie Chen, Yu Zhang, and Qiang Yang. Multi-task learning in natural language processing: An overview, 2021.

[3] Carlos Coello and M.S. Lechuga, 'Mopso: A proposal for multiple objective particle swarm optimization', volume 2, pp. 1051 – 1056, (02 2002).

[4] Kalyanmoy Deb, *Multi-objective Optimization*, 403–449, 01 2014.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, (2016).

[6] Andrzej Jaszkiewicz, 'Multiple objective metaheuristic algorithms for combinatorial optimization', (2001).

[7] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma, 'Fairness-aware classifier with prejudice remover regularizer', in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, volume 7524 LNAI, pp. 35–50. Springer, Springer, Berlin, Heidelberg, (2012).

[8] James Kennedy, *Particle Swarm Optimization*, 760–766, Springer US, Boston, MA, 2010.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[10] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, 'Deep learning face attributes in the wild', in *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, (2015).

[11] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al., *Microeconomic theory*, volume 1, Oxford university press New York, 1995.

[12] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan, 'A survey on bias and fairness in machine learning', *ACM Comput. Surv.*, **54**(6), (jul 2021).

[13] Idel Montalvo, Joaquín Izquierdo, Rafael Pérez-García, and Manuel Herrera, 'Improved performance of pso with self-adaptive parameters for computing the optimal design of water supply systems', *Engineering Applications of Artificial Intelligence*, **23**(5), 727–735, (2010). Advances in metaheuristics for hard optimization: new trends and case studies.

[14] Arjun Roy and Eirini Ntoutsi, 'Learning to teach fairness-aware deep multi-task learning', in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part I*, pp. 710–726. Springer, (2023).

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., 'Imagenet large scale visual recognition challenge', *International journal of computer vision*, **115**, 211–252, (2015).

[16] Antonio Torralba and Alexei A. Efros, 'Unbiased look at dataset bias', in *CVPR 2011*, pp. 1521–1528, (2011).

[17] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool, 'Multi-task learning for dense prediction tasks: A survey', *IEEE transactions on pattern analysis and machine intelligence*, **44**(7), 3614–3633, (2021).

[18] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky, 'Towards fairness in visual recognition: Effective strategies for bias mitigation', in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8919–8928, (2020).

[19] Yu Zhang and Qiang Yang, 'A survey on multi-task learning', *IEEE Transactions on Knowledge and Data Engineering*, **34**(12), 5586–5609, (2021).