TRANSFER LEARNING BASED FACIAL GENDER DETECTION OF GITHUB PROFILES

Minaoar Hossain Tanzil

minaoar.tanzil@ucalgary.ca
Department of Electrical & Software Engineering
University of Calgary

ABSTRACT

Gender detection is a crucial part of diversity research in software engineering. As one of the largest open-source software hosting site, GitHub hosts millions of public repositories contributed by millions of developers. Though GitHub stores many important attributes of these developer profiles such as name, email, location, but it does not store or provide gender data. In this research, we attempt to detect the gender of the users from their profile pictures. Instead of training millions of gender-labeled images, we utilize the EfficientNetB0 architecture trained on ImageNet data set and apply machine learning for fine tuning the gender detection. The generated model performed with 88% accuracy on test data and this was applied to real GitHub user profiles for detecting their gender.

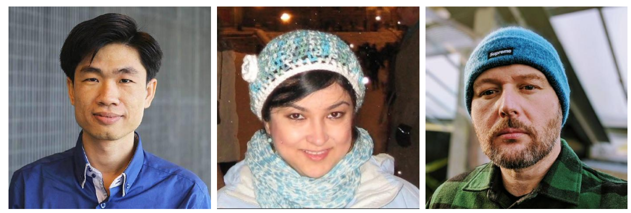
***Index Terms*—** Gender detection, Face detection, Transfer learning, EfficientNet, GitHub, ImageNet

1. INTRODUCTION

GitHub is a large source code host with more than 48M public repositories, 80M users, and billions of commits [1]. GitHub is a matter of interest for software engineering researchers for quite some time; for example, since the mining challenge at the International Conference of Mining Software Repositories, there have been more than thousands of research papers titled GitHub [2]. Besides the technical research topics, a significant researches are also carried on social engineering aspect of collaborative open source software development such as exploring social network of developers, social nature of collaboration [3] etc.

Among these social engineering analyses, gender diversity remains an important topic [4]. But GitHub does not store gender information in its site or provide through the REST API [5]. In order to perform gender based diversity, acceptance, or productivity analysis, researchers have been using mostly name-based searches along with the location or nationality information[6]. There has been also linking between GitHub's registered user address and other social networking site to identify the gender[7]. Even with this cross-linking among GitHub and other sites (Google+), less than 36% of users' gender could be identified [7].

Thanks to Dr. Roberto Medeiros de Souza for continuous guidance.



(a) csokun_chorn (b) sali_1982 (c) dan

Fig. 1. GitHub profiles without gender-deciding name. (a) csokun_chorn's gender cannot be found using name-search database, (b) sali_1982 did not register their name in GitHub, (c) dan's name is treated as unisex by a commercially available name database [8]

Another approach of gender detection could be usage of name search and image detection both as performed by Bin and Alexander [8]. They analyzed on Stack Overflow that more than 50 percent of profile pictures represent human image. Bin and Alexander also recommended that image analysis in addition to name search provides more accurate result in gender detection. The motivation for this research is to identify gender from profile pictures of GitHub users so that the overall detection success can be increased in coordination with the name-only search.

In this regard, the profiles in Figure 1 demonstrate the limitation of gender guessing from name. Here a commercial name search database [9] is used to guess gender from their names. However, if image analysis could be used to detect the gender of these three users, the gender could be detected with higher confidence. This research attempts to improve gender guessing of GitHub users from their image. As gender detection is a form of image classification, there could be numerous methods of implementing the detection procedure. The following section discusses related gender detection techniques in the literature and why transfer learning with CNN architecture was chosen.

2. RELATED WORKS

With comparatively recent surge on online data availability as training dataset, e.g. ImageNet [10], coupled with graphical processing units based computational power, deeper neural networks started to be used in overall image analysis. VGG16 [11] has been a highly regarded pioneer for using convolutional neural networks (CNN) in large scale image analysis. Following the success of VGG16, InceptionV3 [12],ResNet [13] also further enhanced inception module, residual learning for improving image recognition performance. While such models provided incremental improvement on the image classification problems, the EfficientNet reached the top of ImageNet leader-board with their state of the art CNN model and moreover, with the model scaling research [14].

Besides the general classification problem, few researchers proposed their own deep CNNs for face [15] or specifically, gender and age detection [16]. Researchers in [17] proposed their lightweight custom CNN to detect emotion and gender real-time and reported 96% accuracy against IMDB gender dataset inspired by VGG16 [11].

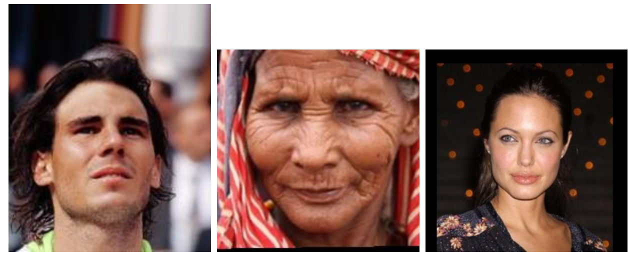
Though the ImageNet challenge models are trained on 14M+ of image dataset, the dataset does not have specific image classes for gender. If we try to develop new models for gender detection from scratch, it would require massive amount of image and enormous GPU power like those models. Instead, we could use the prominent models' weight (such as EfficientNet, VGG, ResNet etc.) calculated on ImageNet dataset and apply custom classification of gender detection over those model weights. This kind of transfer learning has been popular in image classification and also specifically for face detection or recognition problems [18].

In this research, we applied transfer learning [19] technique on top of the high performing architectures such as EfficientNet, VGG, and ResNet, and also on MobileNet which is a family of light-weight model suitable for mobile or embedded system [20]. In the following section we elaborate the detailed methodology of using transfer learning on a limited gender labeled dataset to classify the profile genders.

3. MATERIALS AND METHODS

3.1. Dataset

For training the model (including validation), two separate datasets were used and based on the performance, one training dataset was finally selected. For the testing, a different dataset was used in all models. Table 1 provides the count of female and male images in each dataset. Initially, the UTKFace dataset was used as training dataset, since it had image for age ranging 1 to 116 years and also contained ethnic diversity. Finally, the CelebA database was used as it provided higher accuracy. Figure 2 shows sample images from each data source.



(a) CelebA: Nadal (b) UTK: 84years (c) Kaggle: Female

Fig. 2. Sample images from the training (a,b) and test (c) datasets

Dataset Name	Purpose	Male	Female	Total
UTKFace [21]	Training	12,391	11,314	23,705
CelebA [22]	Training	84,434	118,164	202,598
Kaggle [23]	Testing	17,678	9,489	27,167

Table 1. Dataset size and source

Besides the dataset for model training and testing, a final dataset of 18,412 GitHub profile images was prepared. The detailed process of developing this image dataset is covered in the methods subsection.

3.2. Methods

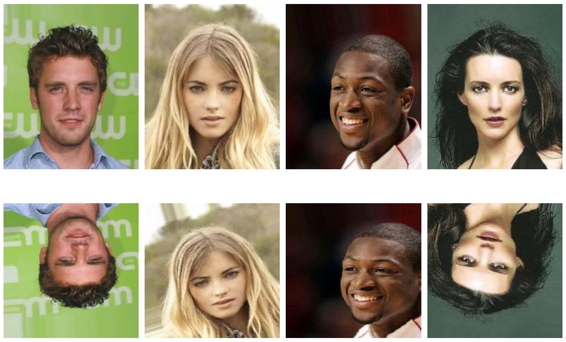
3.2.1. Data Preparation and Augmentation

The CelebA dataset came with a csv file along with the facial attributes and all images in a single folder. A script was prepared to classify the images according to the gender and make two folders of Male and Female. All images were fed to data train generator; and 80% of data was selected as training dataset and rest 20% was used for validation. The Kaggle dataset came in separate male and female folders and hence did not require any preprocessing. Those were directly used as input to the testing model.

All of training, validation, and testing images were augmented by random rotation of up to 20 degrees. In addition, vertical and horizontal mirroring was randomized, shear angle intensity was applied up to 20%. The width and height of the images were shifted up to 20% and the remaining area was filled with the values of the nearest data points, as depicted in Figure 3.

The target dataset of GitHub profile pictures was collected by developing a data scraping engine for GitHub. Firstly, using the GitHub REST API [5], profile picture of 47,438 users were collected; few samples shown in Figure 4. In order to filter face-only images, OpenCV's HaarCascade frontal face detection was used [24]. After the face detection, 18,412 images (around 39% of 47,438 collected profiles) were prepared as final set of target image set.

(a) Original Images



(b) Augmented Images

Fig. 3. Augmentation applied to training and test datasets

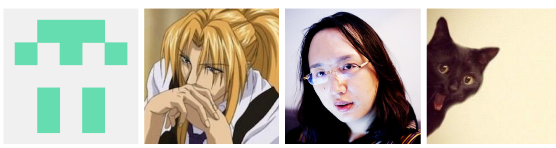


Fig. 4. Types of user profile images in GitHub: (a) GitHub default avatar, (b) Computer-generated Images, (c) Face photo, (d) Other non-face image

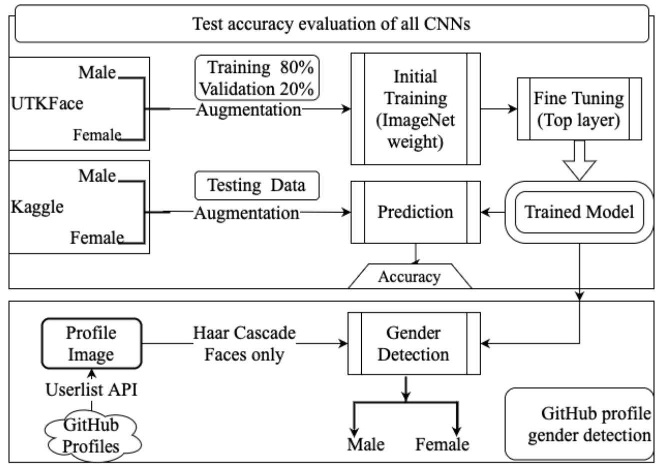


Fig. 5. Flow chart of the experimental setup

(a) Male: 0.84 (b) Female: 1.0 (c) Male: 0.99 (d) Female: 1.0

Fig. 6. Profile pictures with gender probability

3.2.2. *Training and Validation*

For training the image dataset, prominent CNNs are taken as base model trained with the ImageNet data [10]. The top convolution, flattening, and activation layer is stripped off and on top of this base model, a custom final layer is added which flattens the data, and provides the activation layer with target classes (in this case male and female gender).

In the experiments, Adam optimizer is used with initial learning rate of 0.0001 and the learning rate is halved in every tenth iteration of the training. In the initial training phase, the base model is not trainable, and up to 100 epochs have been used; and in the final fine tuning phase, all layers of the model is trained in up to 50 epochs. In every phase, if the validation accuracy does not improve within 20 consecutive epochs, the training stops.

In the first phase of this research, all selected CNNs (EfficientNetB0, VGG16, VGG19, ResNet50, Resnet50v2, MobileNet, MobileNetV2) were trained with UTKFace [21] training data (23K images) and accuracy performance was evaluated using the test dataset (27K images) from Kaggle [23]. All these output models were also used for detecting the GitHub profiles. After this initial screening phase, only the CNN with highest accuracy was further trained and evaluated with the CelebA dataset (202K images) [22]. This final model was again used to detect the target dataset from the GitHub user profiles for generating the improved results.

4. RESULTS AND DISCUSSION

4.1. Results

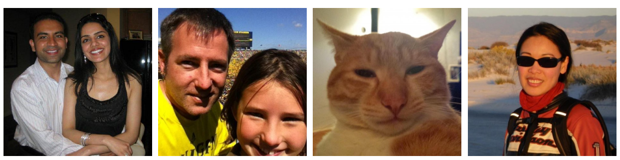
The complete training and testing was run for all seven CNNs with same training set (UTKFace data) and test set (Kaggle data). As can be seen from Table 2, EfficientNetB0 outperforms other CNNs by a a large margin - EfficientNetB0 has a test accuracy of 78.4% whereas the closest accuracy was achieved by VGG16 with 69.7%. Every model was also used to detect the gender of the real GitHub user profiles. While most of the models detected female profiles in the range between 15% to 34%, ResNet50V2 and MobileNetV2 exceptionally detected less that 2% of female images.

4.2. Discussion

First interesting observation from this experiment is the performance of EfficientNet architecture. There was remarkable difference in accuracy of EfficientNet compared with other models. The base model of EfficientNet comprises of MBConvNet, a variant of MobileNet family; still MobileNet and MobileNetV2 had least accuracy among all seven architectures.

CNN	Train Size	Test Size	Test Accuracy	Detection Result			
				Female	Male	Female%	Total
EfficientNetB0	23,705	27,167	**78.4%**	3,654	14,758	19.8%	18,412
VGG16	23,705	27,167	69.7%	5,513	12,899	29.9%	18,412
VGG19	23,705	27,167	67.9%	5,006	13,406	27.2%	18,412
Resnet50	23,705	27,167	68.2%	2,786	15,626	15.1%	18,412
Resnet50V2	23,705	27,167	65.1%	41	18,371	0.2%	18,412
MobileNet	23,705	27,167	56.6%	6,183	12,229	33.6%	18,412
MobileNetV2	23,705	27,167	64.8%	262	18,150	1.4%	18,412
EfficientNetB0	***202,598**	27,167	**88.1%**	3,042	15,370	16.5%	18,412

Table 2. Training accuracy and detection result for all CNNs. *This model is trained with CelebA dataset.



(a) Male: 0.99 (b) Male: 1.0 (c) Male: 0.63 (d) Male: .99

(e) Female: .99 (f) Female: .99 (g) Female: 1.0 (h) Female: .68

Fig. 7. Wrong (or questionable) gender detection of GitHub profiles

The next noteworthy finding from the results is the accuracy improvement of models by training with different dataset. The same architecture and test setup produced almost 10% more accuracy with CelebA dataset which contained 202K images, compared to the UTKFace dataset which comprised of 23K images.

Besides the higher number of images in the training dataset, there could be another reason for better accuracy with the CelebA dataset. As can be seen from Figure 2c (Angelina Jolie), the images from Kaggle test dataset might also contain higher number of celebrities' images. As a result of such similarity of training and test domain, the accuracy with CelebA trained model can be higher. On the contrary, the initial training dataset from UTKFace contained more common people from varying age and ethnic background.

Finally, the most important observation is the detection accuracy of the trained model. Though the test accuracy of 88% is commendable, with the real-world GitHub user profiles, there was a good amount of wrong detection as can be seen from the Figure 7. When there are multiple faces of male and female, the model tends to select male gender with high probability. Also there is mixed output for animal faces, a cat is detected as male and a dog was detected as female. This also points out the limitation on how HaarCascade works to detect the 'face' in an image. As there were 18K images detected by the model, it was not possible to calculate the exact accuracy of detection. But my manual observation of 100 samples from each gender, it was observed that male detection accuracy is around 97%, whereas more than 30% images detected as female actually seem to belong to males.

5. CONCLUSION AND FUTURE WORKS

In spite of the achieved accuracy of 88% on the test sets by the current research, there can be further improvements as observed by concerns listed in the discussion section.

The test dataset could be improved by using a more balanced dataset such as UTKFace so that the domain similarity between training and test dataset becomes lower.

In order to detect images containing real human (not avatar, animal, or other wild images), there can be another deep learning model instead of the OpenCV HaarCascade.

As a large number (30%) of females are wrongly detected, there should be further investigation on root cause and the architecture and/or training dataset to improve the detection accuracy.

Finally, the creation of a gender annotated image database with such a large engineering community of 80M users can also be a potentially significant research project.

6. REFERENCES

[1] GitHub, *Public Repositories and Users in GitHub*, https://github.com/search?q=is:public, 2021, Last accessed on 04 December 2021.

[2] Google Scholar, *Scholarly Articles Titled GitHub since 2014*, https://bit.ly/3DtBkqm, 2021, Last accessed on 04 December 2021.

[3] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge," in *2013 International Conference on Social Computing*, 2013, pp. 188–195.

[4] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark GJ van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov, "Gender and tenure diversity in github teams," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 3789–3798.

[5] GitHub, *GitHub REST API Reference for Users*, https://docs.github.com/en/rest/reference/users, 2021, Last accessed on 04 December 2021.

[6] Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik, "Gender, Representation and Online Participation: A Quantitative Study," *Interacting with Computers*, vol. 26, no. 5, pp. 488–511, 09 2013.

[7] Josh Terrell, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson Murphy-Hill, Chris Parnin, and Jon Stallings, "Gender differences and bias in open source: Pull request acceptance of women versus men," *PeerJ Computer Science*, vol. 3, pp. e111, 2017.

[8] Bin Lin and Alexander Serebrenik, "Recognizing gender of stack overflow users," in *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*, 2016, pp. 425–429.

[9] Gender Checker, *Check Gender by Name Search*, https://genderchecker.com/pages/search-engine, 2021, Last accessed on 05 December 2021.

[10] 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*, vol. 115, no. 3, pp. 211–252, 2015.

[11] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[13] 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*, 2016, pp. 770–778.

[14] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[15] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu, "Face detection, bounding box aggregation and pose estimation for robust facial landmark localisation in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 160–169.

[16] Gil Levi and Tal Hassner, "Age and gender classification using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 34–42.

[17] Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger, "Real-time convolutional neural networks for emotion and gender classification," *arXiv preprint arXiv:1710.07557*, 2017.

[18] Xudong Cao, David Wipf, Fang Wen, Genquan Duan, and Jian Sun, "A practical transfer learning algorithm for face verification," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3208–3215.

[19] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[20] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[21] Song Yang Zhang, Zhifei and Hairong Qi, "Age progression/regression by conditional adversarial autoencoder," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[23] Kaggle, *Biggest gender/face recognition dataset*, https://www.kaggle.com/maciejgronczynski/biggest-genderface-recognition-dataset, 2021, Last accessed on 05 December 2021.

[24] Rainer Lienhart and Jochen Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings. international conference on image processing*. IEEE, 2002, vol. 1, pp. I–I.