Real-world Image Recognition for Multiple Human Attributes

School of Computing, National University of Singapore

Niu Yunpeng
niuyunpeng@u.nus.edu

Wang Debang
debang@u.nus.edu

Yu Pei, Henry
e0032343@u.nus.edu

Daniel Koh Chong Xiang
e0004105@u.nus.edu

Goh Wen Zhong, William
e0004779@u.nus.edu

Tan Ying Lin, Felicia
e0035673@u.nus.edu

ABSTRACT

Human attributes recognition is the infrastructure of human re-identification systems. Although there has been extensive amount of research in classification of single attribute such as gender and age, it is still an open topic for a feasible approach of multiple human attributes recognition. This subject becomes more exciting in real-world unconstrained scenarios. We study some state-of-the-art works in this area and design several Convolutional Neural Network (CNN) models to recognize two attributes, gender and long/short sleeves. We also present a generic framework for multiple attributes recognition on single object.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; *Supervised learning by classification*;

KEYWORDS

Convolutional Neural Network, human re-identification, image recognition

1 INTRODUCTION

Nowadays, we do not lack data of human beings in image or video format. In Singapore, the government has installed more than 80,000 police cameras. In China, people have used the camera system to develop the "City Brain" project. These CCTV cameras have provided the authorities with the ability to capture almost every action happening at any corner. However, it becomes practically impossible to manually watch all the video recordings and understand what is happening/has happened, such as re-identify the appearance of a target person. It would be useful if machine can help us summarize the key attributes of humans' actions in these videos, whose result could then be easily supplied to humans or other tools to get an overall picture and perform higher-level analysis effectively (since the amount of data has decreased tremendously).

Recent advances in computer vision research could help us achieve this goal. New algorithms have increased the accuracy of models to extract information and gain high-level understanding from an image. Convolutional Neural Network (CNN) is one popular class of models, most commonly used in image and video recognition. CNN is highly successful in these fields due to their capability of automatic feature extraction (and thus dimensionality reduction) and their ability to correlate information in localised regions.

When attempting to uniquely identify a human, usually we do not look only for a single attribute such as gender, but combination of several key attributes. Similarly, in order to develop a machine learning model for human re-identification, we have to empower the model with ability to recognize multiple attributes together. To train such a model using supervised learning, it is necessary to obtain a large-scale dataset with annotations of multiple attributes as well. Thanks to the previous work [9], the WIDER attribute dataset is available for us to use.

We begin with reseatching on some approaches proposed in preliminary studies [2, 6, 10, 15] about recognition of common human attributes, primarily regarding CNN models. We then develop a few CNN models with similar architecture on two different datasets. To cater for our tight time constraints, we limit the scope of the models and focus on 2 specific human attributes, namely gender and long/short sleeves. Then, several methods are explored to augment our models, such as regularization and dropout.

2 RELATED WORK

Handling non-square images. Data preprocessing is an important step in training CNNs using images with a wide variety of aspect ratios, especially when the image may contain irrelevant information to the subject of interest. For example, when performing classification on human emotions [13], it is beneficial to perform face detection. Similarly, when performing classification tasks based on human body-parts, such as human pose estimation, using a sliding window detector can improve the performance of the model [8]. Feature scores obtained across all of the windows are aggregated to give a final score, which is used in classification. However, there are some drawbacks to the above-mentioned methods. Facial detection and image segmentation requires labelled data to train a separate model, and using a sliding window results in significantly more computations and lower performance speeds. In this project, we attempt to train models without external models to perform segmentation, omitting the reliance on annotated data of various human body-parts as well as pre-trained models.

Pixel data distribution. Often in Computer Vision problems, models need to deal with images across many different contexts. This results in images with varying illumination and possibly other environmental factors. One of the more established methods is performing batch normalization on the training data. This method not only addresses variations in environmental influence on the image, but also addresses the issue of huge variation on the borders of images. A comparative study of batch normalization on the CIFAR-10 dataset [17] shows the performance increase on 3 CNNs with significantly different architectures. Other methods have also been to deal with varying illumination, such as Zero Component Analysis [11]. In this project, we did not apply any of these preprocessing techniques to deal with varying illumination in images, so that we can focus on experimentation and fine-tuning of the model architecture and regularization methods. This area will be explored in future work of the project.

Model architecture. One of our goals of exploring classification of different human attributes using a shared pool of features is to investigate correlation of different human attributes. In the process of designing the experiment, we tested a few CNN architectures and evaluated their effectiveness of correlating features to classify 2 specific human attributes: gender and whether they are wearing long sleeve shirts. Before delving into finer details, it is important that we fit our model complexity to the data that we are using. We know from [7] that increasing both the width and depth of a CNN can significantly increase its accuracy in classifying an image. However, a CNN with too much modeling capacity can easily overfit when given insufficient quantity and variety in data. In this project, we seek to build a simple model and sacrifice the accuracy at an acceptable level.

3 METHOD

We built two models for the gender classifier (G) and the long/short-sleeve classifier (L/S), whose diagram is affiliated in Figure 1 and 2.[1] The input of our models are resized images of heads for G and full body for L/F. Our models consists of 4 convolutional neural layers and 2 fully-connected layers,mostly with ReLU as the activation function. The loss functions of our models are sigmoid cross entropy loss. The two models have slightly different implementations w.r.t. the loss function, but are effectively the same. We use stochastic gradient descent (SGD) with momentum for G and Adam optimizer for L/S. We use different optimizers to explore more possibilities. The different optimizers can be of course unified given more time for fine-tuning.

To increase the accuracy and boost training speed, we experimented a few techniques as follows.

- **Dropout layers:** It has significant improvement in reducing overfitting and increasing total learning speed. We have set it to 0.4 in our models. More discussions are in 5.1.3.
- **Overlapping pooling:** We changed pooling parameters to have stride smaller than kernel size to overlap the pooling layers. Therefore adjacent pooling window will overlap and more options are available. It proves to have some improvements for L/S model.
- **Local response normalization (LSR):** We inserted LSR after pooling layer for L/S model. LSR was first used in AlexNet [7] to improve the generalizability of the model. LSR introduces lateral inhibition through competition among neural signal strengths. However, the improvement is minimal.
- L1/L2 regularization: We tried L1/L2 regularization in the convolutional kernel but got little improvement.
- **Data augmentation:** Data augmentation methods are good for multiplying the dataset, thus increasing accuracy. Some data augmentation methods we have tried are flipping, translating, rotating and salt-and-pepper noise.
- **Apply filter:** Most pictures have background that contains no information for the task. And different pictures have different position and portion of background. We tried to use CNN to filter out the background, but it didn't contribute a lot to the accuracy.

Since our network topology is rather small, most regularization techniques will not be too effective.

The accuracy can be further improved by stacking more convolution layers and adding more channels. However, the training time, fine-tuning time and prediction time will all increase. And the requirement for hardware, especially RAM, will curb our application from wide-spread application. Thus, we decide to stay simple and sacrifice the accuracy at an acceptable level.

We built our model using TensorFlow [1] and PyTorch [12] libraries. Tensorflow has more helper functions, more supports and more popularity. But PyTorch has a more sensible logic flow, more concise syntax and easier to use.

4 EVALUATION

4.1 Dataset

We have used two different datasets to train and evaluate the different models that we have designed, the WIDER attribute dataset [9] and the LFW face dataset [5].

The WIDER dataset is a large-scale dataset of human images in unconstrained settings. It included 13789 pictures, in total of which containing 57524 humans. As presented in the work by Li et al. [9], the large-scale annotation on a wide range of human attributes makes it stand out from the previous datasets, such as the HAT dataset [14]. This makes the WIDER dataset uniquely important for us as part of the objective is to explore the approach to classify multiple attributes from a single object (i.e., a human image in this case).

The LFW (Labeled Faces in the Wild) dataset [5] is another well-studied dataset of over 13000 human facial images. This dataset contains a sufficient amount of high-quality images (i.e., cropped to only include facial part, scaled to the same size). Although the images are only labelled with gender, it provides a good starting point for us to build a basic CNN model.

4.2 Baseline

There have been extensive researches done in the field of classification of human attributes from images such as gender, age, etc. Different methods have been explored to improve the performance of the classification model, most popular of which are Support Vector Machine (SVM) and Convolutional Neural Network (CNN). In this paper, we have decided to discuss about the classification of two key attributes, gender and long sleeves. Such consideration makes it concise but enough to support our discussion later on the general framework of classification of multiple attributes from a single object.

Table 1 cited the results of gender prediction on the LFW dataset and long sleeve prediction on the WIDER dataset from a few state-of-the-art works. Since previous works have achieved a relatively high level of accuracy, it is hard and also irrelevant to the objective of this project to boost the accuracy even up. However, this does not defeat the significance of this project since we could build a model with a similar level of accuracy and focus on approaches to

[1] A Git repository containing all related code and instructions is available online at https://github.com/yunpengn/MultiAttrCNN.

Table 1: Performance Comparison with Baselines

Method	Gender	Long Sleeves
Random Guess	50%	50%
SVM	88.96% [15]	N/A
CNN	96.86% [6]	86.0% [9]
Our Works	89.5%	80%

classification of multiple attributes together, which is still an open topic.

4.3 Experiment Results

We have built a few Convolutional Neural Network (CNN) models of different topologies to classify gender or long sleeves on images of human beings. They have been trained and validated on either the WIDER dataset or the LFW dataset. Their performance has been compared and the differences of the performance are analysed against the nature of the two datasets.

4.3.1 Gender classification for the WIDER dataset. We started with building a gender classification model for the WIDER dataset using CNNs. Later, we would refer to this network as *Model A*.

We first extracted persons from the contexts by cropping the images to given bounding boxes in the annotations. Since these cropped parts are of different size, we need to transform (resize) them to a uniform size before feeding them to the CNN model. However, the width-to-height ratios of these parts could still be different. There are two methods to fix this problem: 1) pad the image with pixels of single color; 2) stretch the image using interpolation. We have attempted both approaches and found the latter leads to a better result. This is because, in the former approach, the padded pixels of single color would mislead the model.

We built a preliminary model with the following topology and got an accuracy of 71%.

4.3.2 Gender classification for the LFW dataset. Based on the model mentioned in Section 4.1.1 and the previous work by Antipov, Berrani, and Dugelay [2], we have designed and refined a model with the topology shown in Figure 1. Later, we would refer to this network as *Model B*.

As discussed in [2], although CNNs are the primary choice for most computer vision tasks today, it suffers from the problem of too high computational and memory requirements. Thus, to make our model practically useful, we have to keep it simple but yet accurate enough.

We trained and evaluated this model using a subset of LFW dataset, consisting of around 6000 images. This subset was further randomly divided into two parts, the training set (80%) and the validation set (20%).

With 25 epochs, the training could be completed in about 25 minutes on a i5-7360U processor without GPU support. We have performed the training and validation a few times and eventually achieved an accuracy of about 89.5%.

After getting the model, we performed some real-world testing. We created a tiny dataset of around 30 images, most of which are sampled from pictures of student volunteers from the National University of Singapore (NUS). The accuracy is not as high as the one measured in the validation dataset. More analysis on this can be found in Section 5.2.2.

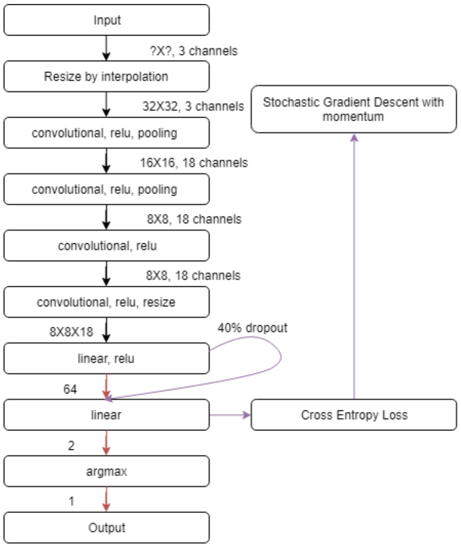


Figure 1: Topology for Gender Classifier

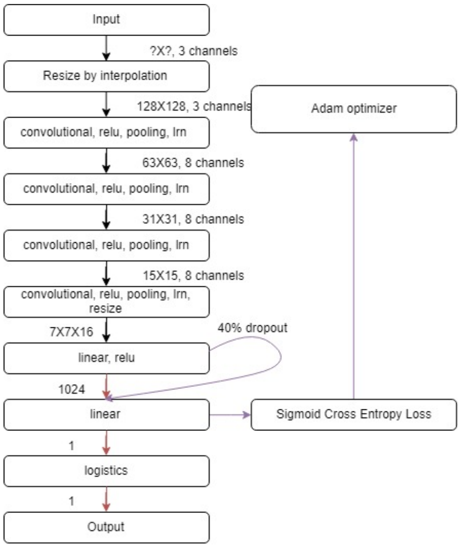


Figure 2: Topology for Long-sleeves Classifier

4.3.3 Long-sleeves classification for the WIDER dataset. Similarly, we have also developed a CNN model with the topology shown in Figure 2 and got an accuracy of about 80%. Later, we would refer to this network as *Model C*.

We extracted individual people using the same method as in 4.3.1. In order to utilise all information present in the scene, the original image from which the person was cropped from was also fed into the model through the same CNN architecture. To simulate segmentation of the image, a mask was computed through

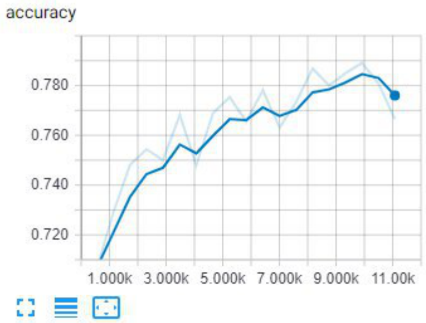


Figure 3: Accuracy for Long-sleeves Classifier

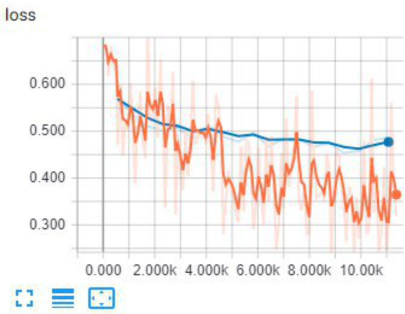


Figure 4: Loss for Long-sleeves Classifier

3 repetitive convolutions on the cropped image with a kernel size of 5x5. The number of filters for the 3 convolution layers were 8, 8 and 1. This mask is then multiplied with the cropped image before being fed into the CNN model. The model was trained for roughly 11,000 steps with batch size of 32 using all the data available from the WIDER attribute dataset [9]. The training was done on a GeForce 940MX for roughly 30 minutes. Some statistics for the training process of *Model C* are shown in Figure 3 and 4.

To further improve the model, a few data augmentation methods were utilised, such as random image flipping, change in brightness of the image, cropping a subset of the image, rotating the image and adding salt and pepper noise. After some testing, the best combination for the model was random flipping and salt and pepper noise, resulting in an improvement from 75% to 80% accuracy. The final model uses the architecture as shown in Figure 2.

5 DISCUSSION

5.1 Parameter Tuning of CNN models

We have performed the following microscopic analysis on the CNN models that we have built and thus tuned some parameters of the network to improve the accuracy of the model.

5.1.1 Overfitting. Since neural network (and especially the philosophy of deep learning) does not limit the number of layers, theoretically we can represent a function of infinitely large order and thus can learn anything. Although this sounds nice, this idea makes our network in general prone to overfitting.

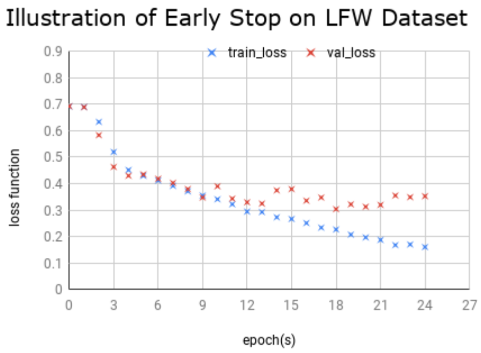


Figure 5: Illustration of Early Stop

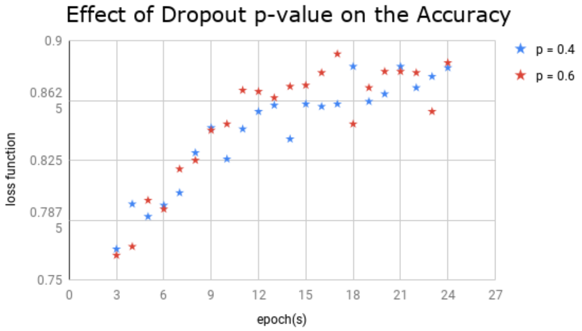


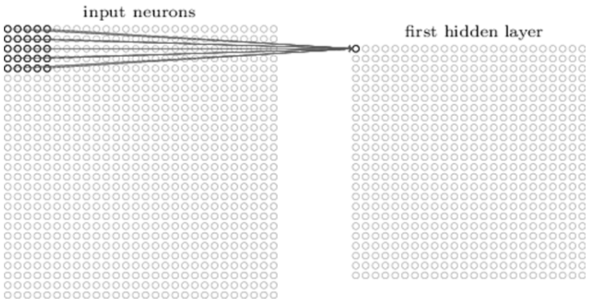
Figure 6: Illustration of Different Dropout p-values

From the figure above, we can see that the training loss keeps decreasing after every iteration of training. However, the validation loss will not go down anymore after a certain number of epochs. This shows that the generalization ability of the model does not improve further after sometime. Even if we can see improvement in the training loss, it is not meaningful since such improvement is due to features specific to the data points in the training set.

5.1.2 Early Stop. As mentioned in the last section, if there is no more improvement to validation loss, it is useless to continue training even if the training loss keeps decreasing. Thus, we can apply a technique called early stopping. Popular open-source machine libraries such as TensorFlow [1] and PyTorch [12] have support for it. In Figure 5, we should stop after 12 epochs.

5.1.3 Dropout p-value. Regularization is a standard cure for overfitting. Apart from the traditional regularization techniques such as L1 and L2 regularization, adding dropout layers [16] is also useful and has been used widely. However, the p-value of dropout layer could be tricky to decide. In Figure 6, the network performs better when $p = 0.6$.

5.1.4 Data augmentation. During the training of *Model C*, some data augmentation methods were used. However, using all of the methods resulted in the model being unable to converge. This could be due to the model's inability to handle rotations and cropping, as these augmentations obscure some of the information available, or



Visualization of 5 x 5 filter convolving around an input volume and producing an activation map

Figure 7: Illustration of Different Dropout p-values

introduces patches of black pixels (in Tensorflow's implementation of image rotation). However, methods such as image flipping and adding of salt and pepper noise improved the accuracy of the model. Image flipping along the horizontal axis is likely to be beneficial since the model should be invariant to such changes in orientation, as it does not matter if a man faces leftwards or rightwards. Similarly for salt and pepper noise, it distorts high frequency noise and edges present in the image, encouraging the model to ignore smaller details such as every edge and instead focus on lower frequency details such as the shape of an object, like someone's arm or torso. These 2 augmentations help the model to generalise better by being resistant to image flipping and high frequency noise.

5.2 Comparison between Three Models

In Section 4.3, we described three models and listed the experiment results on them. Below we would like to analyse the causes of the differences in their performance.

5.2.1 Extraction of Region of Interest. Comparing *Model B* with *Model A*, we observed a significant improvement. This is mainly due to the nature of the two dataset and the way we processed the input images.

In the WIDER dataset, after we cropped the image according to the bounding boxes given, we get images of the whole human body. However, in the LFW dataset, we directly get images of the facial part only. It is obvious that most information used to classify the gender from a human image is from the face (i.e., it is relatively hard to predict the gender from the clothes or the body shape). Thus, for the images from the WIDER dataset, the body and the background of the person other than the facial part effectively becomes noise and could mislead the gender classifier. Since the facial part would also be visually smaller (and thus represented by fewer pixels), the image carries less useful information as well.

In addition, without extracting the facial part of human images, the pictures are not "aligned" and thus the kernel in the convolutional layers could not extract features easily.

Figure 7 by [3] visualizes how the convolutional layers in CNN work. It applies a kernel on a region called receptive field. The kernel is moved like a sliding window to iterate through all pixels in the picture. If all images have been cropped to only include the facial part, we can approximately state that the center of the image would be the nose of the person. However, for images extracted from the WIDER dataset, the center of each image would vary. For some images, it could be the hand of the person; for some other images, it could be even the background. Thus, we find that it would be hard for the kernels to learn useful features when passing over a certain receptive field if that region is not "aligned" and does not carry any particular type of information.

Table 2: Different Ethnicity Groups in the LFW Dataset

Ethnicity	Count
Black	1122
White	11045
Asian	1063
Unknown	3

The above analysis suggests that it is helpful to crop the input image and only keep a certain area before feeding it to the CNN model. We call such a region **Region of Interest (RoI)**. For instance, the **RoI** for gender classifier would be the face; while the RoI for long-sleeves classifier would be the upper body.

5.2.2 Ethnicity Imbalance in the LFW Dataset. As mentioned in Section 4.3.2, we have observed a performance degrade when the model trained on the LFW dataset was tested on a tiny dataset we collected from NUS students.

After some research, we found some statistics from [4]. As shown in Table 2, there is a significant imbalance of ethnicity in the LFW dataset. Since the dataset of NUS students we collected mainly consists of Asians, the profile of the humans in the images would be very different from the LFW dataset as LFW mainly consists of white people. This could be one of the causes of decrease in performance. While the dataset has a higher proportion of white people than the others, we can relieve the problem by giving higher weights to misclassified black and Asian people.

5.3 Framework for Multiple Attributes Classification on a Single Object

Given the experiments and discussion above, we would like to propose a framework for classification of multiple key attributes on a single object (such as images of human beings). The framework for multiple human attributes recognition is illustrated in Figure 8, while a generic version of the framework is presented in Figure 9.

In the generic framework, a raw input image is defined as a *scenario*, which contains one or many *object*(s). For instance, given a *scenario* of crowd of people, it contains many people *object*s. The key *attribute*s of each person could belong to the same or different *RoI*(s). For instance, the *RoI* for gender and age would both be face, while gender and long-sleeves would have different *RoI*s.

6 CONCLUSION

In conclusion, we have built a light-weight convolutional neural network model for gender and long/short sleeves classification, with accuracy that is satisfactory for practical use. We did not achieve the state-of-art accuracy for both classifications, which is a trade-off made for the entire system.

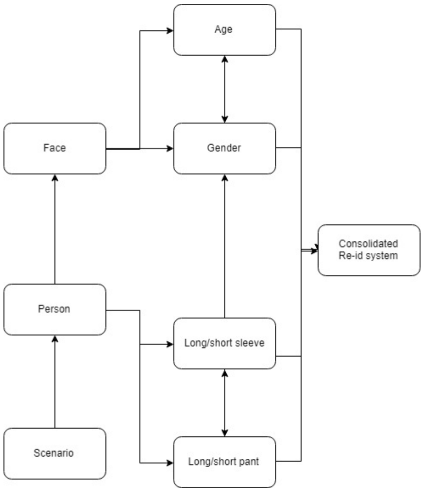


Figure 8: Workflow of the Framework for Human Multiple Attributes Classification

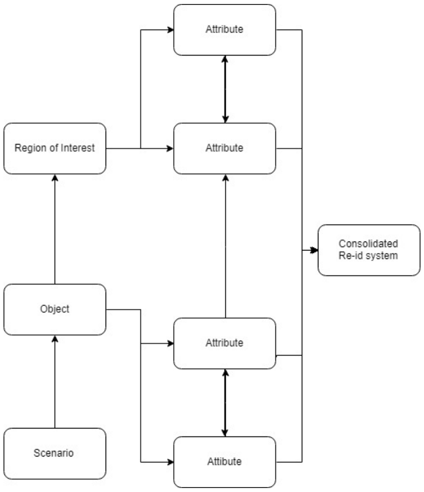


Figure 9: Workflow of the Generic Framework for Multiple Attributes Classification

An area that deserves further investigation in future work is how batch normalization, Zero Component Analysis and other preprocessing steps that deal with image illumination and other environmental factors.

In addition, we proposed a generic framework for multiple attributes classification on single object. Due to time and manpower constraint, we are unable to provide a production-ready system for multiple human attributes recognition. Nevertheless, we have trained several CNN models for genders and long/short sleeves as an example. Other part of the re-identification system can be extended easily following the framework proposed in the future.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[2] Grigory Antipov, Sid-Ahmed Berrani, and Jean-Luc Dugelay. 2016. Minimalistic CNN-based ensemble model for gender prediction from face images. *Pattern Recognition Letters* 70 (2016), 59–65.

[3] A. Deshpande. 2016. Beginner's Guide To Understand Convolutional Neural Networks. (2016). https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

[4] Hu Han and Anil K Jain. 2014. Age, gender and race estimation from unconstrained face images. *Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, MSU Tech. Rep.(MSU-CSE-14-5)* (2014).

[5] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition.*

[6] Sen Jia and Nello Cristianini. 2015. Learning to classify gender from four million images. *Pattern Recognition Letters* 58 (2015), 35–41.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[8] Sijin Li, Zhi-Qiang Liu, and Antoni B Chan. 2014. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 482–489.

[9] Yining Li, Chen Huang, Chen C. Loy, and Xiaoou Tang. 2016. Human attribute recognition by deep hierarchical contexts, Vol. 9910. 684–700.

[10] Hong Liu, Yuan Gao, and Can Wang. 2014. Gender identification in unconstrained scenarios using Self-Similarity of Gradients features. IEEE, 5911–5915.

[11] Kuntal Kumar Pal and KS Sudeep. 2016. Preprocessing for image classification by convolutional neural networks. In *Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE International Conference on.* IEEE, 1778–1781.

[12] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[13] Diah Anggraeni Pitaloka, Ajeng Wulandari, T Basaruddin, and Dewi Yanti Liliana. 2017. Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition. *Procedia Computer Science* 116 (2017), 523–529.

[14] Gaurav Sharma and Frederic Jurie. 2011. Learning discriminative spatial representation for image classification. In *BMVC 2011-British Machine Vision Conference.* BMVA Press, 1–11.

[15] Huang-Chia Shih. 2013. Robust gender classification using a precise patch histogram. *Pattern Recognition* 46, 2 (2013), 519–528.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.

[17] Vignesh Thakkar, Suman Tewary, and Chandan Chakraborty. 2018. Batch Normalization in Convolutional Neural NetworksâĂŤA comparative study with CIFAR-10 data. In *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT).* IEEE, 1–5.