# Ethnicity Detection via Deep Learning

Robert J. Weld, *Student, Author*
Dr. Ghulam Rasool, *Machine Learning Professor*
GitHub Repository Available At: https://github.com/rjweld21/Weld_EthnicityDetection

***Abstract*—Deep learning is currently leading in cutting-edge predictive modeling. Within this paper several preprocessing methods and artificial neural network models are presented toward the goal of detecting the ethnicity of a subject within an image. This ethnicity detection can later be used as feedback into other deep learning models for the prediction of facial aesthetic feature grades and hopefully increase current accuracies of these grading models. Although only VGG16-based and ResNet-based models are created for this project, other famous model architectures were considered to fulfill the need for ethnicity prediction. In the end, models with accuracies up to 94% testing accuracy were produced but future work is still planned to be done.**

***Index Terms*—VGG16, Residual Network (ResNet), GoogLeNet, Inception-V3**

## I. Introduction

THE ethnicity of a person may influence many other factors when attempting to predict or grade other facial aesthetic features. Therefore, the detection of ethnicity may be able to aid other machine learning models whose purpose is to grade facial aesthetic features of a person's face within an image. For this project, the classification of four ethnicities using a model based off of VGG16 and a model based off of ResNet was done. Only four ethnicities were classified for this project because previous data acquired, comprised of over 70,000 images of 1,443 subjects, and used for training only included labeled subject images with ethnicities of African American, Asian, Caucasian and Hispanic. These ethnicity predictions, when used as an input to other models for facial aesthetic feature grading, could improve prediction accuracy of other models.

This project for ethnicity detection was thought up between the author, Robert Weld, and his managers at the L'Oreal USA Technology Incubator. Although there are some resources available online for ethnicity detection, no previous projects were used as reference or influenced the creation of the project presented in this report. In fact, it was only until after the models were created, trained and tested that an online search was done in an attempt to find any previous work toward the same goal.

Some previous work which influenced this project were the artificial neural network architectures for VGG16 [1] and ResNet [2] which will be further discussed in the later parts of this paper. In addition to these previously tested model architectures, some code for the ResNet creation was also based off of problem set 2, question 2, supplied to students in Dr. Rasool's machine learning class. One last piece of previous work that enabled the creation of this project was dlib facial detection and mapping [3]. This library was used for the preprocessing of data.

The data used for training, validation and testing of models within this project is owned by L'Oreal and cannot be disclosed to the public. Some information that can be disclosed is that all the data collected was done so in a study conducted in various cities across the United States. From this study, 1,443 subjects had their pictures taken to account for just under 14,000 images within the data set. Subjects also had videos taken of them turning their face from looking center, to looking left, then turning their face around to look right before returning their face back to center. From these videos, frames were extracted at various time points to account for another 60,000 images within the data set.

Future work for this project will include the collection of additional data to add to the dataset with more ethnicities and regional data as well. In essence, this regional data will allow for possibly creating a model which detects ethnicity and nationality. These additional prediction details could again further the prediction accuracy of the facial feature grading. These new models will most likely be trained by using transfer learning with the weights derived from the training of the models in this report.

## II. Methods

As previously stated, the use of a VGG16-based and ResNet model were used for ethnicity prediction, but these were not the only models considered. The GoogLeNet [4] and Inception-V3 [5] models were also considered but were later determined not to be the first models tested. These models were chosen to be implemented last simply so the creator and author could test other models, such as VGG16 and ResNet, and see their performance first. If all else failed with VGG16 and ResNet based models, then inception modules could be implemented and the model would hopefully be able to decide which are the best convolutional layers to use within each module of the model. Luckily, as it will be later discussed, performance with the VGG16 and ResNet based models was good enough that a need for inception modules was not deemed necessary.

Before creating any models, however, it was known that the data would need to be preprocessed to practically normalize some aspects of the image data. This preprocessing was done with the dlib library, written in C++ but implemented in python, which allowed for the mapping of certain points on the face within each subject image. The mapped points from dlib on each face would outline the jawline, nose, mouth, eyes and eyebrows of a face which was enough data to enable masking, cropping and resizing of the faces. With the jawline and eyebrow points used as reference, forehead points were interpolated and checked on subjects. Although creating one method, for all subjects, of interpolating forehead points would not perfectly fit each subject, a robust method which mapped forehead points well for most subjects was created and implemented into the facial cropping preprocessing script. Once all the facial points were mapped, the leftmost, rightmost, uppermost and lowermost points were taken as reference to create a bounding box for cropping.

In the first iteration of preprocessing, the facial outline found with dlib jawline points and interpolated forehead points was kept as is in the original image while the rest of the image was filled with zeros on every channel. This was later found to be an ineffective preprocessing method for the faces between each image were in different locations and also of differing sizes. This motivated the creator to alter the facial cropping script so that the images would include facial masking and be zero-filled just as before but now the facial bounding box would have its inner contents cropped from the semi-processed image, resized to 224x224 and then saved as a new image. Although this second iteration of preprocessing increased accuracy during training, it was not sufficient enough to be acceptable as a final training accuracy to further test.

This lead to a third and final iteration of preprocessing which produced acceptable accuracies during training. The only difference made between the second and third iterations of preprocessing was that no facial masking was used in the third iteration. It is believed that this promoted accuracy because, although there may be some background in the image, the jawline is more discrete. In addition to this, each subject's neck may deduce better skin color detection leading to better ethnicity detection. An example of an original image can be found below in Figure 1 and an example of each preprocessing method can be found below in Figure 2.

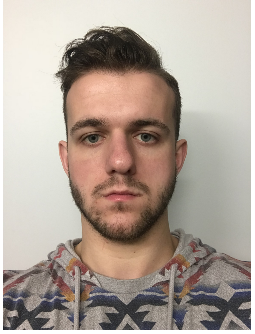

Figure 1: Original Image Example

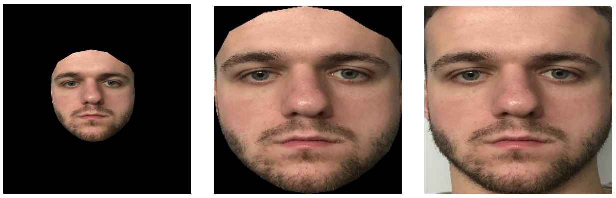

Figure 2: From left to right, iterations 1, 2 and 3 of preprocessing executed on Figure 1

With the images processed to be input to deep learning models, a first model to test out was now to be created. It had been discussed with L'Oreal machine learning employees which model to try first and eventually VGG16 was chosen. This model was chosen because of its robustness for image classification, especially if transfer learning with ImageNet weights is implemented. The full summary of this architecture can be found in the iPython notebook file on the GitHub repository and also as an image in the repository. In this summary it is seen that the model ends up with over 15 million total parameters. Overall, for this model, the VGG16 model from Keras was loaded with ImageNet weights, excluded the top fully connected layers and was set to accept inputs of shape 224x224x3. The top fully connected layers were excluded to substantially reduce the amount of parameters within the model and therefore decrease processing times during training and testing. The input was also set to 224x224x3 because this is the size used to train the original VGG16 model and the size that all of the preprocessed images were resized to. From here, the output of this model base was set to feed into a global average pooling layer which then fed into a fully connected dense layer and lastly into another fully connected layer with softmax activation. The global average pooling layer was added to the model because it is believed this layer will encourage the model to identify aspects from the whole image as described in [6]. The fully connected layers were then also used to eventually output predictions, using softmax probability, into one of four classes. The amount of epochs to train on was fixed to 50 with a batch size of 32 for all iterations of hyperparameter tuning. This model was trained many times and the accuracies from the first couple of training iterations are what motivated the preprocessing method evolution previously described. The last aspect of this model to note is that stochastic gradient decent (SGD) was used as an optimizer. This optimization method was chosen because it is the least likely of optimization methods to cause over fitting. With the ImageNet weights, there is a much higher probability that the model can overfit when compared to a naive architecture.

Once the VGG16 model had been trained and tested, it was expected that an architecture with less parameters should be sought out. Not only to lower prediction time, but also because there may be other model architectures which could gain higher prediction accuracies after training. With this motivation, a ResNet model was created based off of some code supplied in problem set 2 of the author's machine learning course he was currently enrolled in taught by Dr. Ghulam Rasool. Originally, the code to create the architecture in problem set 2 was to accept input shapes much smaller than 224x224. Now, the ResNet model for this project was altered to include more stacks to accept and fully process the larger input shape of 224x224. The new model resonates a shape of 7x7x512 at the output of the sixth stack which is then fed into a global average pooling layer, with the same intent as it was used in the VGG16 model. This then fed into a flattening layer and eventually another dense layer with softmax activation output for the four ethnicity classes. This model included many more layers and also had less than one-third of the parameters compared to the VGG16-based model with just under 5 million total parameters. Again the amount of epochs for training was set to 50 and this time the batch size was increased slightly to 128. As for the optimizer, SGD was again used. In the first iteration of training for this model, Adam optimization was used but after more research online about ResNet models it was found that SGD is a common optimizer for ResNet models. The switch from Adam to SGD did increase overall accuracy in the end as well. The training of this model then took about 13 hours for the 50 epochs.

## III. Results

### A. VGG16-Based Model Results

In the beginning iterations of training, accuracies around 40% were achieved which sparked the evolution of preprocessing from method 1 to method 2. After the revamping of the preprocessing method, accuracies rose a little higher to around 45%-55%. This was still unacceptable and prediction results were analyzed to find that the model had been vastly overfitting to predict a Caucasian output. It was not originally thought to be done, but after looking back at the amount of examples for each ethnicity it was found that Caucasian examples outnumbered all other individual examples by 140% or more. A reduction function was then created and was used to reduce the amount of Caucasian examples in the dataset by 40%. This reduction in the Caucasian example count now put the amount of Caucasian examples to be slightly less than the Hispanic example count and much closer to the African-American and Asian example counts than previously found.

With this change in the dataset, the model now reached accuracies around 60%-70% but this was still not sufficient and again sparked more motivation to move from preprocessing method 2 to preprocessing method 3. Now, on the first training iteration after moving to preprocessing method 3, an accuracy on the lower end of 80% was achieved. After some optimization of the learning rate, a training accuracy of 96% and validation accuracy of 93% was achieved leading to a testing accuracy of 94%. The two learning rates found to lead to these accuracies were values of 0.0005 and 0.0009. Accuracy and loss metric plots can be found below in Figure 3. Here, the traces labeled with ”1” in the legend refer to the model with learning rate 0.0005 and the traces labeled with ”2” refer to the model with learning rate 0.0009. Overall, this model took around 30 hours to complete all 50 epochs when trained on one NVIDIA Tesla M60 GPU. Luckily, two of these GPUs were able to be used at once which allowed for more rapid training and optimization.

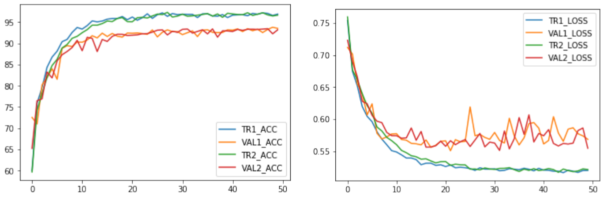

Figure 3: VGG16-Based Model Metrics

### B. ResNet-Based Model Results

After all data preprocessing issues had been ironed out during the VGG16 model creation, the ResNet model was able to be developed with acceptable accuracies in much less time. With the ResNet architecture previously described, and trained as a naive model, more promising results were acquired for ethnicity detection. The first iteration of training quickly overfit to achieve training accuracies around 90% and validation accuracies around 60%. Here, training was interrupted and the learning rate for SGD was modified. The learning rate was changed to have a learning rate schedule of 0.015 for epochs 0-40 then reduce the learning rate to 0.010 for epochs 41-50. This produced much better results, which can be seen plotted below in Figure 4. The model did end up overfitting slightly more than the VGG16 model created but was still able to achieve 91% validation accuracy and 96% training accuracy. This model was then run on the testing set and achieved 91% accuracy again.

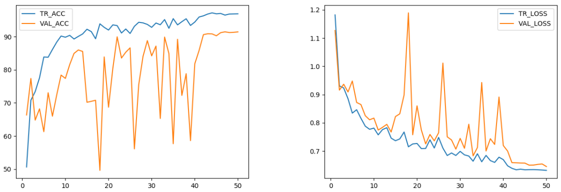

Figure 4: ResNet-Based Model Metrics

Considering the large fluctuation of validation accuracy and loss for this ResNet model, further training iterations were done to see if a more stable model could be created with hyperparameter tuning. Unfortunately, with more iterations and changed hyperparameters, only more unstable models with validation accuracies up to 86% were found.

## IV. Discussion

With the results found, a lot was learned between the preprocessing and actual training of models. For the preprocessing, it is believed that the first method attempted failed at generating anything near acceptable results because of the varying locations of the faces. With the masked out faces being in different places for each example, the model most likely focused on location of each face more rather than the features within each face. Since there were many more images of Caucasian subjects during training with the first method of preprocessing, the model simply predicted Caucasian for each image and arrived at accuracies around 40%.

Moving from preprocessing method one to two, although it increased accuracy slightly, was also found to be inferior when compared to using images processed with method three for training. Method two was thought to yield less accurate prediction model this method eliminated part of the face and neck on every image. In some cases, hair was also found to be masked out which could also lead to better detection. With more hyperparameter tuning it is believed that a model of around 80% may be possible to achieve, but it is not worth the time cost to train such a model just to try and find the best possible results that this preprocessing method can lead to.

While changing the preprocessing method to iteration three may have helped increase accuracy of the models, dropping the Caucasian example size to closer match the example sizes of the other three classes was found to be very necessary for increasing accuracy. In the early iterations of training, metrics were output to answer the question, ”if the model mispredicts a label, what is the misprediction vs the actual label?” From this metric it was found that the Caucasian class was overwhelmingly predicted more often than any other class. It was now clear that the models were being overfit to predict Caucasian for the images. Counter acting this class example difference by reducing the Caucasian example size was very influential in achieving higher prediction accuracies.

After training the models, it was found that the VGG16-based model was much more stable and yielded better accuracies. This architecture was also trained on many more iterations than the ResNet model though and also implemented transfer learning with the ImageNet weights. For future work on this project, the ResNet model will be further tuned in an attempt to create a more stable model. If steadier validation results can be generated and achieve the same current testing accuracy around 91% then the prediction time decrease between the VGG16 model and the ResNet model may be worth the slight accuracy dip. A couple of training iterations using ImageNet weights for the ResNet model may also be tried to see if a steadier model can be created with transfer learning.

## V. Conclusion

In conclusion, the goal of this project was to detect ethnicity of a subject within an image. Multiple preprocessing and artificial neural network models were tried to achieve testing results around 94% and 91% between the differing models. Preprocessing was found to be very influential for achieving these acceptable accuracies. The VGG16-based model was found to be very robust and achieve the highest testing accuracy but had the most amount of total parameters leading to the highest processing cost. The ResNet model created still achieved a very acceptable accuracy of 91% with one-third the parameters of the VGG16 model but seemed quite unstable during training and validation. Future work is planned to be done on the ResNet model in attempts to possibly raise testing accuracy and make the model more stable.

## References

[1] K. Simonyan and A. Zisserman, ”Very Deep Convolutional Networks for Large-Scale Image Recognition”, Computing Research Repository Journal, vol. 1409.1556, 2014.

[2] K. He, X. Zhang, S. Ren and J. Sun, ”Deep Residual Learning for Image Recognition”, Computing Research Repository Journal, vol. 1512.03385, 2015.

[3] D. King, ”Dlib-ml: A Machine Learning Toolkit”, Journal of Machine Learning Research, vol. 10, no. 1755-1758, 2009.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, ”Going Deeper with Convolutions”, Computing Research Repository Journal, vol. 1409.4842, 2014.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, ”Rethinking the Inception Architecture for Computer Vision”, Computing Research Repository Journal, vol. 1512.00567, 2015.

[6] M. Oquab, L. Bottou, I. Laptev and J. Sivic, ”Is object localization for free? weakly-supervised learning with convolutional neural networks”, Computer Vision and Pattern Recognition (CVPR), vol. 10, no. 1109, 2015.