FACIAL FEATURE DETECTION USING VARIOUS DEEP LEARNING MODELS

Progress Report

Author(s) Name(s):Subin Varghese, Whitney Torres

Author CougarNetID(s):1361427, 2040223

Summary

Our goals for this project are to explore different deep learning models and their ability to perform image classification on our Facial Feature and Attribute dataset. We want to be able to identify specific attributes of human faces like gender, hair color, or facial hair. To do this, we chose to use the CelebFaces Attributes dataset which has over 200,000 photos with 40 different annotated attributes. To date, we have explored single class models by using the Inception V3 model from Keras and Tensorflow pretrained on ImageNet. From our base model that we created, we used two different optimizers - Stochastic Gradient Descent and Adam - and compared the accuracy of the two. When using the Gender attribute data over 15 epochs, we were able to achieve a testing accuracy over 90%. Between the two runs, we were able to see that the model achieved a higher test and training accuracy using the Adam optimizer. For the Smiling attribute, although it was only run for 1 epoch, the same result was also seen when running the two different optimizer models. Moving forward on our project, we will want to apply what we have learned so far into being able to create a multilabel classifier. After creating that, one important goal for us is to use the trained model as a method of feature extraction which will be fed into a SVM for prediction instead of a softmax layer.

Introduction

A growing area of interest in deep learning is facial feature detection. This component is continually becoming a more common aspect of daily life, such as facial recognition on phones and computers or social media photo tagging. Inspired by this, we wanted to explore our own capabilities using deep learning methods applied to detecting simple facial features and attributes in images. From these models, we hoped to be able to identify details of someone's face, such as gender, hair color, or the fact if they are smiling or not. To achieve this, a variety of different deep learning models will be created and compared based on their ability to accurately assess the attributes we choose to be most interested in. This method of comparing different models allows us to explore and learn about different networks and their application to our datasets.

Datasets

The dataset chosen for this project is the "CelebFaces Attributes (CelebA) Dataset" from the researchers at MMLAB at The Chinese University of Hong Kong. This dataset was found on Kaggle and is rated very highly on the site and used commonly for noncommercial research purposes [1]. CelebA is made of over 200,000 celebrity face photos that are each annotated with 40 attributes. Of these 200,000 photos, there are about 10,000 individual celebrity faces used [1]. Each of the facial attributes is a binary classification to mark either the presence or absence of it in the given photo. In addition to the collection of images, the dataset also provides other accompanying csv documents. This includes a recommended partitioning list, bounding box information, face landmark coordinates, and the attribute labels. The size of this dataset allows us to have reliable training, validation, and testing. Though the data set is large, the variables are not distributed evenly, for example only a small fraction of the dataset has bald individuals, but a large number of individuals with no beard.

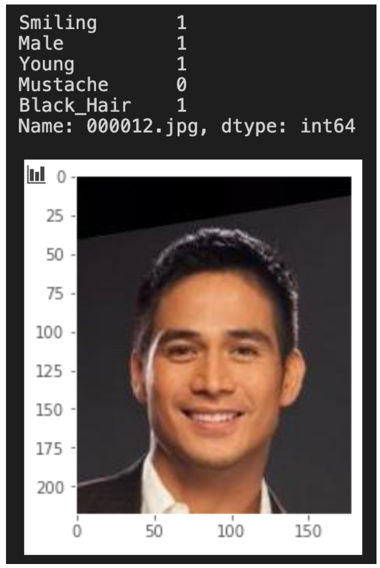

Fig. 1. Dataset Example with 5 attribute annotations

Methods

To date, we have experimented with single class models and their ability to assess an image for gender and smiling. To get to this point, our first decision was to use Tensorflow and Keras and be able to load and read the CelebA dataset. By loading the Celeb Attributes csv using the python library pandas we were able to load the csv into a pandas data frame then easily manipulate the created data frame. This allowed us to easily partition the data into training, testing, and validation as recommended by the dataset authors. The partitioned training data was then augmented with a variety of shifts, zooms, rotations, and other data augmentation methods.

Once the data was ready for use, we chose to use the Keras pre-built Inception V3 model (Fig. 2) that has pretrained weights from ImageNet as a starting point for our model. By using an off-the-shelf-model, we were able to easily reaffirm that our goals for facial attribute detection were attainable. Using the pretrained model allows us to create an accurate model and cut the time needed to train the model.

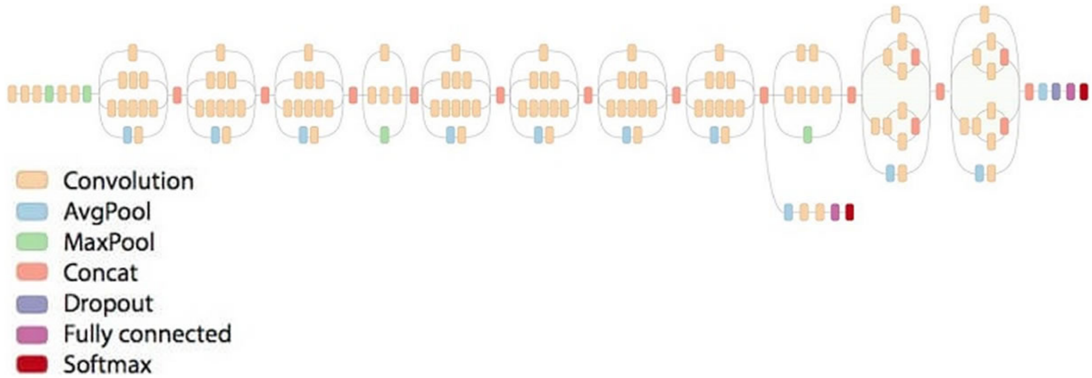

Fig. 2. Inception V3 Architecture. Source: [2]

To test the model with our data, we first used a Stochastic Gradient Descent (SGD) Optimizer to detect Gender over 15 epochs. The results of this were quantified by tracking the Loss Function and the Accuracy of both the training and validation datasets. As a comparison for this model, we tested the Gender detection of the Inception V3 model with Adam as the optimizer. Again, the success of this experiment was quantified via the loss function and accuracy values. We repeated this same process with the Smiling attribute, changing the optimizer between SGD and Adam, but only for 1 epoch. Only one epoch was used in order to see the effect of transfer learning. All of these trained models we created were then used on the testing data to find its testing accuracy and an accompanying F1 score.

Preliminary results

Results of the prediction of a single class in the data set using the Inception V3 architecture pre-trained on ImageNet.

Model 1a: Gender for 15 Epochs in Inception V3 with SGD

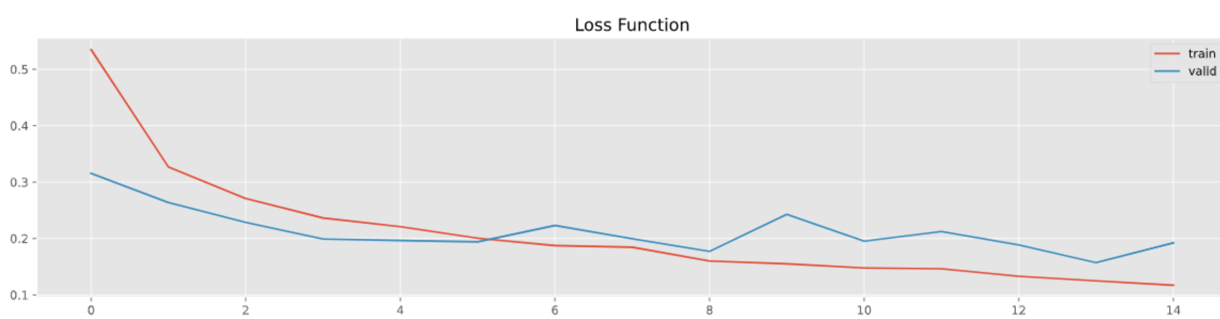

Fig. 3. Loss Function of Model 1a

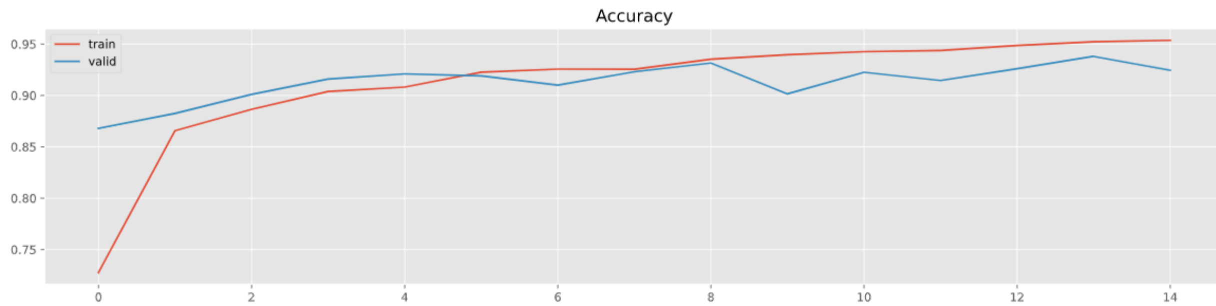

Fig. 4. Accuracy of Model 1a

Table 1
Model 1a Testing Evaluation

Test Accuracy	93.35%
F1 Score	0.9291

Model 2a: Gender for 15 Epochs in Inception V3 with Adam Optimizer

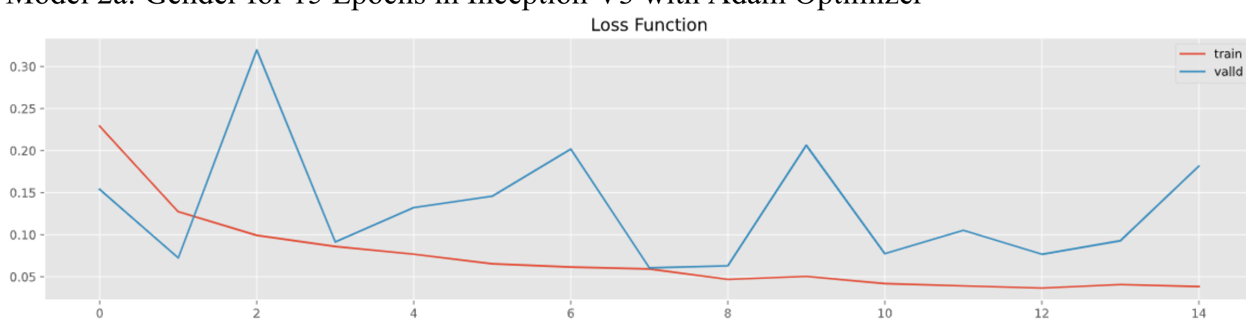

Fig. 5. Loss Function of Model 2a

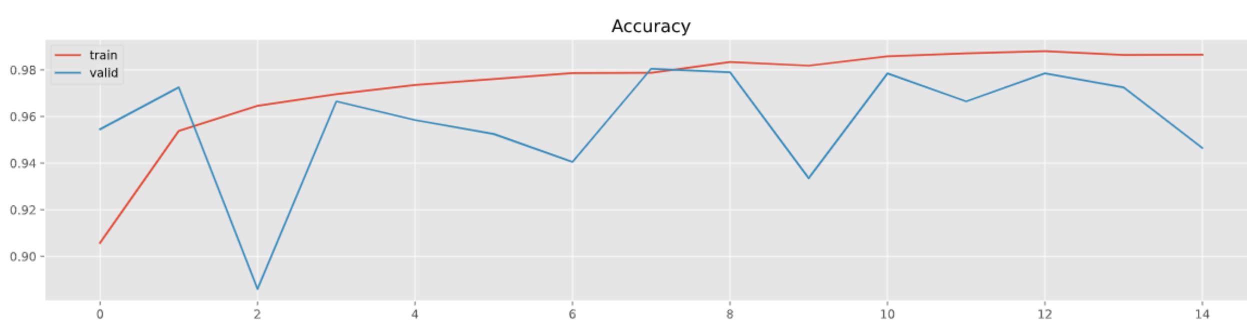

Fig. 6. Accuracy of Model 2a

Table 2
Model 2a Testing Evaluation

Test Accuracy	97.45%
F1 Score	0.97428

Model 1b: Smiling for 1 Epoch in Inception V3 with SGD

Table 3
Model 2a Testing Evaluation

Test Accuracy	72.60%
F1 Score	0.74416

Model 2b: Smiling for 1 Epoch in Inception V3 with Adam Optimizer

Table 4
Model 2b Testing Evaluation

Test Accuracy	80.85%
F1 Score	0.8316

Based on preliminary results, using the Inception V3 model pre-trained on ImageNet then training on the CelebA dataset seems to offer good initial performance. Tables 3 and 4 are the result of training the pre-trained model on the CelebA dataset for only 1 epoch using different optimizers. These results seem to suggest that transfer learning allowed the model to use the patterns extracted from training on ImageNet to be used in CelebA. This allows for the relatively high accuracy from only 1 epoch.

The model was trained to 15 epochs as well using SGD and Adam as the optimizer while using the same learning rate between the two. During the training to 15 epochs, almost no improvements occurred on validation test accuracy after the 10th epoch. Adam was able to reach a higher test accuracy in the same number of epochs as SGD. When comparing the plots of the accuracy shown in Figure 4 and Figure 6, it can be seen that Adam has an erratic validation curve in comparison to the SGD validation curve.

Remaining work and evaluation metrics

As of writing this report, the implementation of SVM is still being worked on. The current plan for implementation for SVM is to use a CNN that we train for multilabel classification and use that as a means of feature extraction[3]. A feature vector will be extracted before the fully connected layers in the CNN and saved for every observation. The SVM will then be trained on the extracted features. We intend to use the scikit learn library to implement the SVM while also trying various builtin kernels for the SVM.

Currently, we are working on changing our code to handle multi-label prediction, unlike the current implementation where only a single classifier is being used. The new model will output a one hot encoded vector of size 40, the number of classes in the set. Also, the code used to obtain preliminary results did not implement the GPU for training, training time was roughly 150 minutes for 15 epochs. We will be updating the code to utilize an RTX 2080 GPU on a local machine. After which we intend to use Amazon SageMaker for further training depending on performance needs.

References

[1] J. Li, "CelebFaces Attributes (CelebA) Dataset," *Kaggle*, 01-Jun-2018. [Online]. Available: https://www.kaggle.com/jessicali9530/celeba-dataset?select=img_align_celeba. [Accessed: 24-Sep-2020].

[2] A. Milton-Barker, "Inception V3 Deep Convolutional Architecture for Classifying Acute Myeloid/Lymphoblastic Leukemia," *Intel*, 17-Feb-2019. [Online]. Available: https://software.intel.com/content/www/us/en/develop/articles/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic.html. [Accessed: 2-Nov-2020]

[3] Alzubaidi L, Fadhel MA, Al-Shamma O, Zhang J, Duan Y. Deep Learning Models for Classification of Red Blood Cells in Microscopy Images to Aid in Sickle Cell Anemia Diagnosis. Electronics. 2020; 9(3):427.

[4] M. Alvarado, "Image Recognition - Gender Detection - Inceptionv3," *Kaggle,* 2018. [Online]. Available: https://www.kaggle.com/bmarcos/image-recognition-gender-detection-inceptionv3. [Accessed: 30-Oct-2020]