STA 141C Final Project: Gender Recognition with Keras

Love Chien Mason Del Rio Amay Kharbanda

Member Contributions

1. Love Chien: Logistic Regression of both 50×50 and 100×100 image sizes for comparison purposes, initial data visualization and exploratory data analysis of image data, report writing.

2. Mason Del Rio: Created Convolutional Neural Networks (with/without Regularization) for 50×50 and 100×100 image sizes, parameter experimentation on CNN, data preparation, data creation and saving code, downscaling of images, testing CNN on outside data, organizing code, report writing on methodology, implementation and results for CNN models and data preparation.

3. Amay Kharbanda: Recurrent Neural Networks for 50×50 image sizes to obtain a reasonable classification accuracy, model testing, and report writing.

Introduction

For our project, we investigate the CelebFaces Attributes (CelebA) dataset which is a public dataset that can be downloaded online from kaggle. This dataset contains 202,599 collected images of faces of both male celebrities and female celebrities. In this data, we have a collection of 40 attributes of each image. These attributes are categorical data and store information about the person in the image, such as if they have black hair or arched eyebrows. The data is divided into a training set (162,770), testing set (19,961) and validation set (19,886) respectively. Each column represents an attribute for each row, which represents each celebrity image. Although this dataset is large, we are able to downscale and greyscale the given images so that processing them and the code would not take an inordinate amount of time. Our goal is to create a model that can accurately distinguish a male person's picture from a female person's picture. We would like to train and test our models to recognize images and obtain high accuracy. We will also optimize our models to avoid long processing times by using effective algorithms with appropriate parameters.

Methodology

As stated previously, our goal in this project is to train a model that would best recognize and categorize a picture of a person's face and determine if the person is male or female. Note that the model is classifying each person's image given that they are not non-binary. The methods that we will use to develop a model are Convolutional Neural Networks (CNN) with learning rates 0.001 (default) and 0.01, Regularization Implementation, Logistic Regression, and Recurrent Neural Networks (RNN). Convolutional Neural Networks are able to take images as inputs and assign emphasis to various objects or aspects in the images and distinguish them from one another. We use a validation set as well, rather than just a training and test set for our CNN in order to avoid overfitting to test data. The CNN will adopt a two layer structure with RELU activation functions for each layer, and a final sigmoid activation layer which we interchange with SOFTMAX for comparison. We also attempt to regularize and modify the model with L1 and L2 regularization to avoid overfitting. We conduct a logistic regression model to compare our multi layer models to a single layer model and see which performed the best. We also have logistic regression at hand for making a quick and simple calibrated model that still has the capabilities of returning high accuracy. Since logistic regression is not particularly meant for image classification or as complex as neural networks, we will use its accuracy to compare with CNN's accuracy. Additionally, we will compare our models with RNN with conditions and layers similar to CNN. We would like to observe the better model for classification of images while keeping in mind that RNN and CNN are used for different purposes. We use the same testing, training, validation split of the data for RNN. Moreover, we will make use of the RELU activation function for each layer and a sigmoid activation layer.

Implementation Details

Prior to performing the aforementioned methods, we used the `matplotlib` and `pandas` packages to perform some exploratory data analysis (EDA) to investigate aspects of the data and attributes. To directly implement the deep learning algorithms, we utilized Python's `sklearn`, `keras`, and `tensorflow` packages. We used greyscale and downscaled images of celebrities in order to build several models to categorize images by gender. We downloaded the data online from kaggle, which contained 1 GB of information mainly including 202,599 RGB images of celebrities in 218 by 178 resolution, and also a csv file `list_attr_celeba.csv` that we will use later for labeling our images as male or female. We used the `cv2` package and `os` package to create a function called `data_creation(x)`, which accepts size value of any kind. For our models, we used the values 50 and 100 to create 50×50 and 100×100 grayscale images to determine if those values decreased accuracy of the models significantly. The `data_creation(x)` function navigates our folder containing the images of the celebrities, and utilizes the argument `cv2.IMREAD_GRAYSCALE` to convert each image to a grayscale `np.array` in order to input into our Convolutional Neural Networks, Recurrent Neural Network and our logistic regression models. It then also adds the corresponding gender labels, -1 for female and 1 for male using the `celebData` variable containing the labels from `list_attr_celeba.csv.` We then reshape each `X,` our features, with the specified image size. After, we change the -1 values indicating female stored in the `y` variable to 0 in order for the labels to work with our models. We then, for each size of image, will save the files to '.npy' file types on our machines, and call them back to avoid computing the data again using the `np.load('')` function, with the string input being the name of the file. These files will be called into the `X` and `Y` variables before each model is trained, in our case we used `'datacelebX.npy'` and `'datacelebY.npy'`, respectively, for 100×100 images and `'datacelebX50.npy'` and `'datacelebY50.npy'` for 50×50 images. This data will be fed into the multiple models, and specifically the CNN and RNN models will be constructed with different batch sizes, epochs and learning rates as experimented with in the code. We then save the models with the `model.save()` function, with `'model'` being replaced with the name of the fitted model specified at the end of each model code chunk, and a string of the directory you would like to save your model inside the parenthesis. We use `keras.models.load_model()` to load back the model in a certain folder. Using `'model.summary()'` we are able to call back the structure of the models specified, the shape of the data through each layer, and the amount of parameters used in each layer. Finally, the accuracy and loss of the models are graphed using the `matplotlib.pyplot` package imported as `plt.` The y-axis indicates the accuracy/loss, and the x-axis represents the number of epochs.

Results and Interpretation

We gained much insight from running preliminary EDA. Taking a look at Figure 1 in the Supplementary Material, we can see that there are more images of females in the dataset than males. This observation implies that most of the features or attributes of the images that the models learn belong to more feminine attributes. Furthermore, Figure 2 shows that a large portion of the dataset consists of images of younger people. Besides suggesting the fact that most celebrities may be on the younger side, this observation also suggests that the models will learn more attributes of people that are younger rather than older.

Before explaining the results, we will talk about the different activation functions used and parameters such as epochs, learning rates and batch sizes. The CNN models for this dataset tended to perform better and faster with learning rates at the default 0.001. At 0.01, the model was becoming more accurate at a slightly slower rate than at 0.001. With batch sizes, anywhere between 500 and 1000 was the sweet spot due to our large dataset. When experimenting with other sizes, the model tended to train poorly with higher loss and lower accuracy. Also, there was a slight decrease in performance from 95.41% to 94.64% when using 50×50 images instead of 100×100, but was still competitive with the 100×100 trained models when tested on pictures not from the dataset. Two models were run before the others, model '10010e' and model '10e', both ran for 10 epochs. The model trained on 100×100 images had validation accuracy of 95.87% and a test accuracy of 96.82% after 10 epochs. The one trained on 50×50 images had validation accuracy of 95.77% and a training accuracy of 96.76%. At epochs 3-4, the validation accuracy and test accuracy were around the same, but as more epochs went on the testing accuracy started to become higher than the validation accuracy. This means we were overfitting the model, and determined that epochs 3-4 would be suitable to avoid overfitting. The 'sigmoid' activation function for the output layer worked much better than the 'relu' activation function or the 'softmax' activation function, due to the fact that our models' purposes are for binary classification as opposed to multi-classification.

The accuracy of our CNN models worked reasonably with data even outside of the training set. We loaded images from random celebrities online and processed them to be inputted into our models. After we tried 20 to 30 images, each image for the Convolutional Neural Network for both models trained on 50×50 images and 100×100 images had very strong results with images that had masculine traits. When given an image of a male, the model has values ranging between 90-99% confidence that it indeed had masculine features. With female images we found, the model is observed to have a range of 10-85% confidence that the image is one with masculine traits. Overall, lower percentages equated to being more likely to be feminine.

For the Convolutional Neural Networks with L1 and L2 regularization, the difference in accuracy and loss in both models are substantial. The CNN model trained with L1 regularization (Lasso Regression) is very inaccurate with test data, due to the sparse nature that L1 regularization introduces to the parameters of the model. The CNN model trained with L2 regularization is much higher in accuracy, yet lower than the Neural Networks we trained without the regularization which is expected. We do not want our model to be too overfit to the data because it will not perform as well on outside data. Therefore, we would use the Convolutional Neural Network model with L2 regularization if applied to the real world, or the model trained with 3 epochs.

We used a logistic regression model as a comparison tool to see how much better or worse CNN is at predicting gender based on the given images. For images of size 50×50, the logistic regression model is able to predict male and female images with approximately a 90.8% accuracy score. For images of size 100×100, the model is able to predict with approximately a 91.4% accuracy. Overall, logistic regression performed well and accurately, even though it is not primarily used as an image classification model. We observed that CNN performs better with discerning images of males, but is not as accurate when classifying images of females. Both methods are useful, but since CNN is a deep neural network specifically suited for image classification, it may be preferred over logistic regression to give the model sufficient complexity.

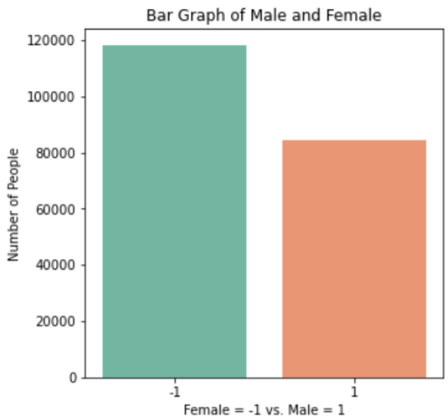
For running the RNN model, we chose to only train the model on the 50×50 image size. We found that 100×100 image size took an excruciatingly long time to run and was not worth the time and computational power. Also, we chose the learning rate to be 0.001. Knowing that a learning rate of 0.01 would increase the computation time for the model, which already has an extremely strenuous training process, we decided not to use the 0.01 learning rate. The batch_size was chosen to be 1000 as a general size, num_epochs was 5 and validation split of 20%. The 'Adam' optimizer was used along with a 'sparse_categorical_crossentropy' loss function. Our model did not train very efficiently due to the high loss. This model had a training accuracy of 86.95%, a validation accuracy of 88.52%, a test accuracy of 88.35%, and a loss of 30.55%. While the accuracy values are not absurdly low, the loss function was extremely high indicating that our model did not train accurately. This was not unexpected as we were using a model not optimized for image classification. This is further corroborated as we analyze sample images to test our model. The model has a high success rate in identifying males however an extremely low success rate for females, with almost every image being identified as a male. Clearly, the RNN is not suitable for this dataset.

Early in our work on the project, we stumbled upon a problem in which loading our data, even in grayscale, was a task that took very long to compute, and would have even taken long for the servers provided for us. We discovered that downscaling our images was the technique that would aid in the problem of long computations for creating our matrices of greyscale converted images. This cut down processing times of creating our data for training and testing, and decreased time in training our models.

The data given to us had 41 attributes listed of each celebrity image, with categories such as “narrow eyes”, “beard”, “receding hairline” that vastly influences the model due to the fact that each of these categories had very few or very high counts. This explains the nature in which the model can identify masculine traits more confidently than feminine, as patterns like baldness and beards can influence the model's parameters. Without RGB images, the model cannot identify lipstick, eyeliner or makeup in general as a very strong indicator of feminine features. Greyscale limits the model to only identifying patterns such as edges, shapes and color from white to black.

We conclude that CNN is the best model for accurately predicting images with masculine features and does indicate it is “aware” that an image is in fact a person with feminine features, as it decreases in percentage that an image is one of a male.

Supplementary Material



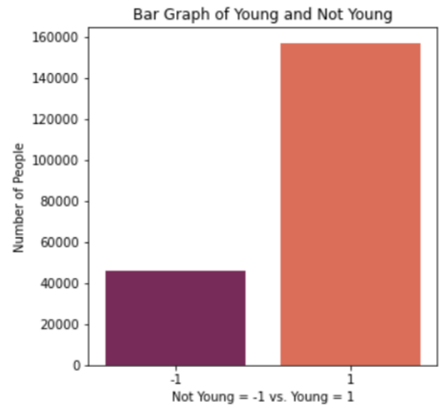
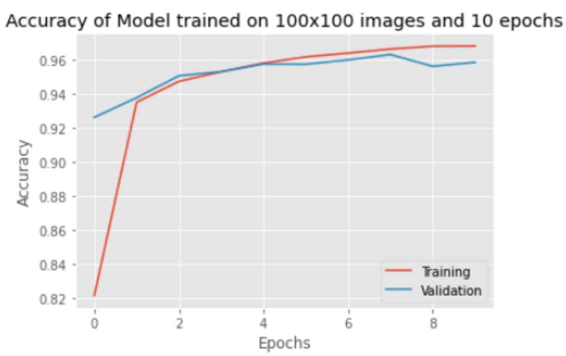
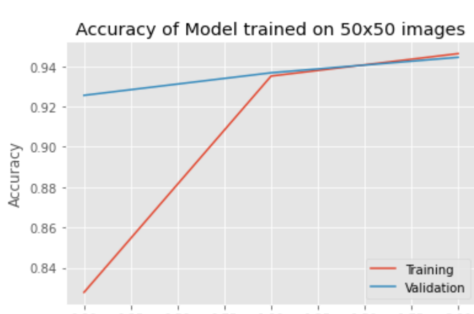
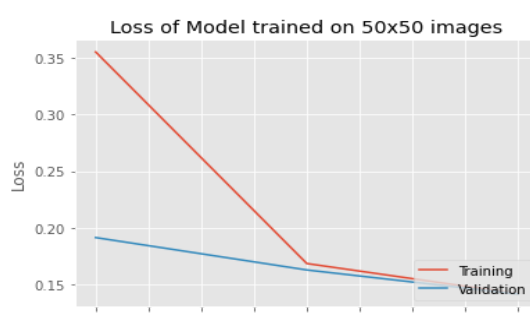


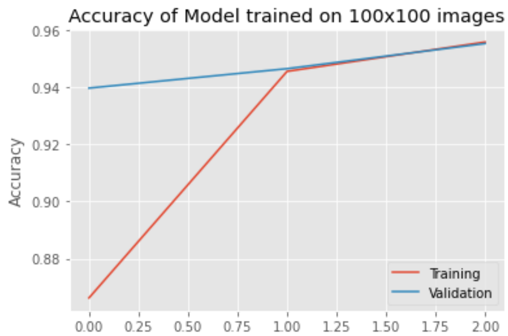
Figure 1.

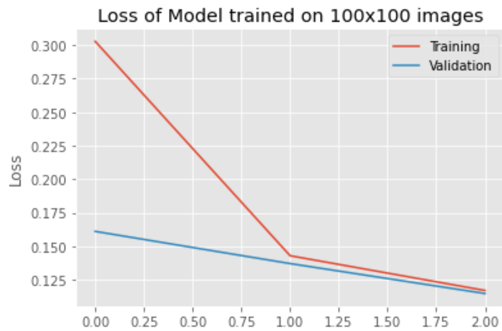
Figure 2.



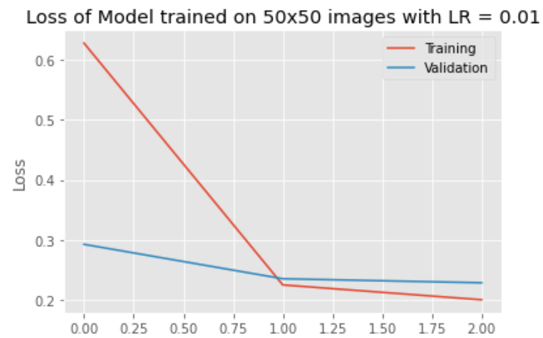


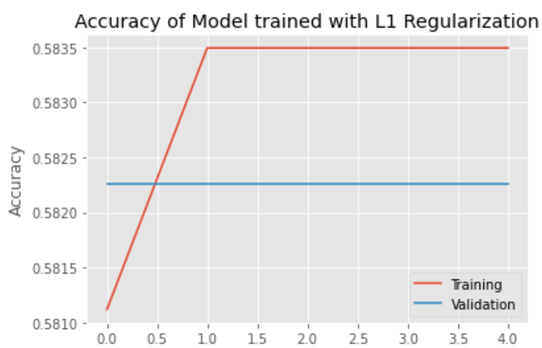


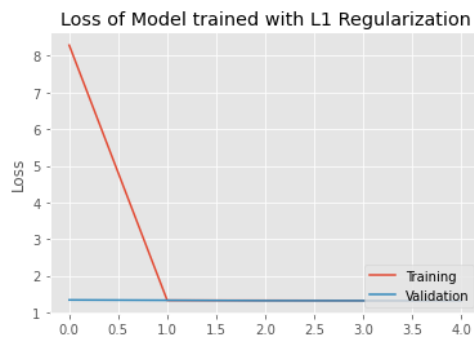

Accuracy of Model trained on 100x100 images
Accuracy
Training
Validation

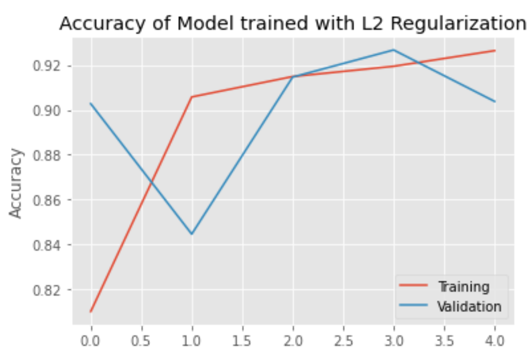

Loss of Model trained on 100x100 images
Loss
Training
Validation


Accuracy of Model trained on 50x50 images and LR = 0.01
Accuracy
Training
Validation

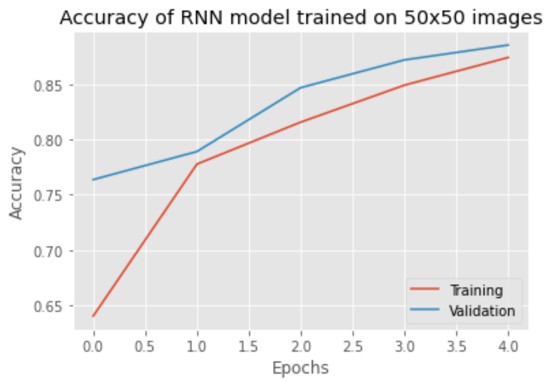

Loss of Model trained on 50x50 images with LR = 0.01
Loss
Training
Validation

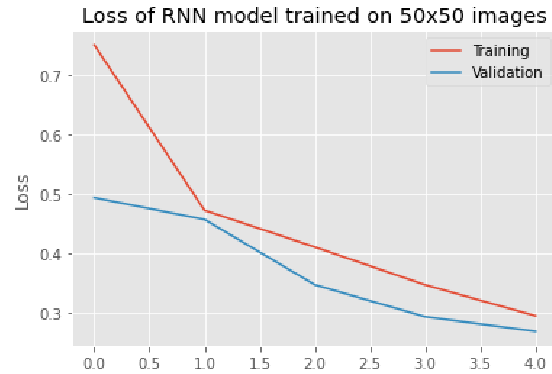

Accuracy of Model trained with L1 Regularization
Accuracy
Training
Validation


Loss of Model trained with L1 Regularization
Loss
Training
Validation

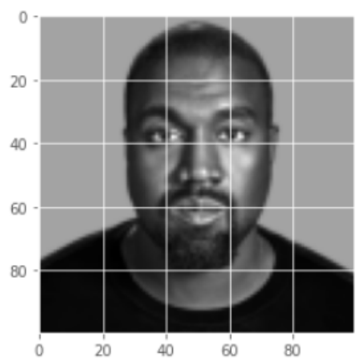

Accuracy of Model trained with L2 Regularization
Accuracy
Training
Validation


Loss of Model trained with L2 Regularization
Loss
Epoch
Training
Validation

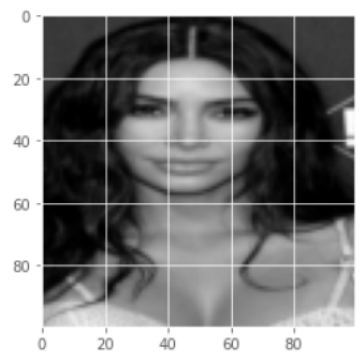




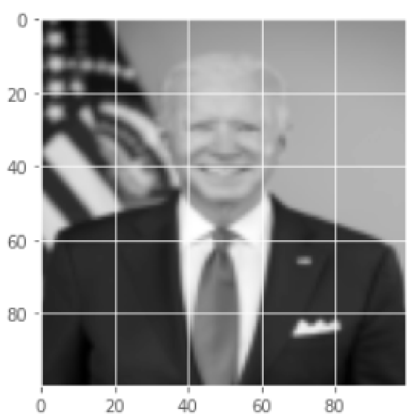
99.59 % likely to be a male



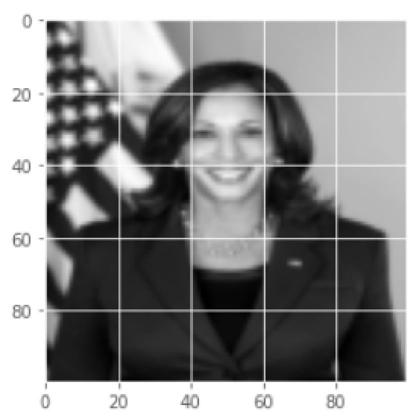
61.48 % likely to be a male



99.87 % likely to be a male



5.1 % likely to be a male



Images of people above are the predictions of the CNN model trained with 100×100 resolution images and 3 epochs.