

**CE/CZ4042 Neural Networks and Deep Learning**

**Gender Classification**

**Group Project Report**

**7th November 2022**

**Chan Ming Kai, Alvin (U1921305J)**

**Jolene Tan (U1921255B)**

**Tay Yu Xuan (U1922631L)**

# **Table Of Contents**

[**Table Of Contents**](#_3jrdkqq1v0c0) **1**

[**List Of Figures**](#_wnad1zj1auem) **2**

[**List Of Tables**](#_b36ka7z2uaua) **3**

[**1 Introduction**](#_8bxvuz3iebs0) **4**

[1.1 Selection of Topic for Course Project](#_p4c3h789hpog) 4

[1.2 Problem Statement](#_1jv56fdaf66z) 4

[1.3 Project Objective and Scope](#_8asdpesed0w8) 4

[**2 Related Works**](#_6fd470o9t17v) **5**

[2.1 VGG16](#_2m4o8vc2eoco) 5

[**3 Methodologies**](#_8t02m6w3akhl) **5**

[3.1 Overview Of Approach](#_nyc60k4m85w4) 5

[**3.2 Data Preprocessing**](#_bzrxu7iflz9c) **5**

[3.2.1 CelebA](#_cyo2px636bz1) 6

[3.2.2 Adience](#_296ia5es0u1a) 6

[3.3 Evaluation Metrics](#_dp5174wazvyy) 6

[3.4 Transfer Learning with VGG16](#_3o72qu83uz89) 6

[3.4.1 Baseline Model Evaluation](#_o18sowwlsxtd) 7

[3.5 Hyperparameters Tuning](#_lax3cmxtci5o) 8

[3.5.1 Hyperparameter Choice](#_1kt379axosv5) 8

[3.5.2 Tuning Approach](#_3a7b2b48khkc) 8

[3.6 Model Selection](#_umxyh0b1cwfq) 8

[3.6.1 Optimal Model Evaluation](#_bux17iv80sus) 9

[**4 Novelties**](#_5u1zx5yi495g) **10**

[4.1 Pre-Training on CelebA and Transfer Learning with Adience](#_ks6joqgi4o3n) 10

[4.1.1 Results Analysis](#_gwuvjej29bsl) 10

[4.2 Feature Importance](#_f2go1ste018l) 10

[4.2.1 Rationale](#_oku7kmlfmqk8) 10

[4.3.2 Approach](#_gddw87li5hfy) 11

[4.3.3 Evaluation](#_a9e0z1w15wr3) 11

[4.3.3.1 Evaluation Metrics](#_m77h94hdi38t) 11

[4.3.3.2 Evaluation of Optimal Model in Predicting Each Feature](#_vq8dc52ftuvp) 12

[4.3.3.3 Evaluation of Feed-Forward Model in Predicting Gender](#_owu2cwcjq0dp) 12

[4.3.3.4 Evaluation of Feature Importance](#_vw2jb9jzhdgu) 12

[**5 Discussion**](#_n977anlx51cb) **13**

[5.1 Discussion On Optimal Model Performance On CelebA Versus Adience](#_urhhu0td088i) 13

[5.2 Discussion On Hyperparameters Tuning Experiments](#_jlqogiyenb9i) 13

[5.3 Discussion On Selection of Features For Feature Importance](#_hdqzsd48xf6t) 13

[**6 References**](#_mka6i8kaoa60) **14**

# **List Of Figures**

| **Figure No.** | **Description Of Figure** |
| --- | --- |
| 1 | VGG16 Architecture |
| 2 | Overview Of Classification Approach |
| 3 | Baseline VGG16-CNN Architecture |
| 4 | Validation Accuracy Of Baseline Model (CelebA |
| 5 | Validation Loss Of Baseline Model (CelebA) |
| 6 | Optimal VGG16-CNN Architecture |
| 7 | Validation Accuracy Of Optimal Model (CelebA) |
| 8 | Validation Loss Of Optimal Model (CelebA) |
| 9 | Validation Accuracy Of Optimal Model (Adience) |
| 10 | Validation Loss Of Optimal Model (Adience) |
| 11 | Overview Of Feature Importance Approach |
| 12 | Force Plot Of SHAP Values |

# **List Of Tables**

| **Table No.** | **Description Of Table** |
| --- | --- |
| 1 | Baseline Model Performance (CelebA) |
| 2 | Hyperparameter Tuning Experiments and Results |
| 3 | Optimal Model Performance (CelebA) |
| 4 | Optimal Model Performance (Adience) |
| 5 | Optimal Model Performance On Other Features |
| 6 | Feed-Forward Neural Network Performance On Gender |
| 7 | Top 3 Most Important Features For Gender Prediction |

# 

# **1 Introduction**

For the CZ4042 Neural Networks and Deep Learning group project, our team is required to propose and execute an application or a research issue that is related to neural networks and deep learning. In particular, we are required to come up with a potential technique for the application or to mitigate the issue, to develop associated codes, and to compare with existing methods.

## **1.1 Selection of Topic for Course Project**

Our team has decided on the topic of gender classification, as it has numerous applications, ranging from law enforcement and marketing intelligence. One use case would be businesses utilising this technology to identify customer segments, understanding what products and services different customers want and can afford. They can then change or develop their marketing strategies and product range accordingly.

## **1.2 Problem Statement**

Gender prediction of a human subject from an image is a binary classification problem with the labels of "male" or "female". Many attempts have been created to solve this issue, however the majority of them were trained on datasets that were not representative of images that such models would be used on, as they were trained on datasets that were strictly controlled and standardised, which may not accurately reflect the domain's actual use cases.

These datasets are made up of frontal-aligned photos that are not obstructed or blurred and obtained in regulated lighting conditions. On the other hand, it is difficult to find photos of people in the wild that have been edited and controlled. Including those found online and on social media platforms. A Dataset with richer representations of human beings in the wild should thus be taken into consideration, as this is the most likely use case for a gender classification model.

## **1.3 Project Objective and Scope**

Our project aims to investigate the best existing architecture for Gender Classification in the wild images.

Our initial approach will consist of utilising existing classification model backbones trained on imagenet, to obtain a baseline value. We will then attempt to improve the performance of this model by modification of hyperparameter tuning. We will also attempt to investigate the effects of multivariate predictions, and the effects on the final accuracy of gender predictions, by performing analysis on output feature importance.

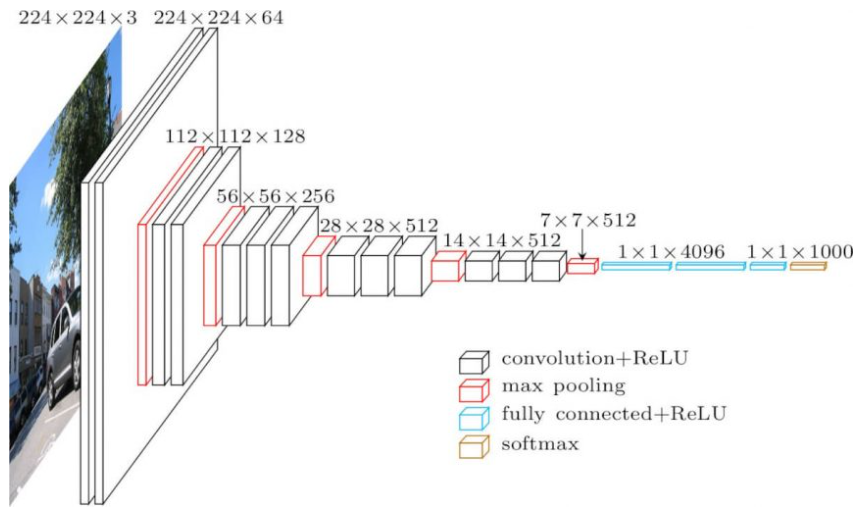
# 

# **2 Related Works**

## **2.1 VGG16**

VGG16 is a 16 layers deep convolutional neural network (CNN) model proposed by Karen Simonyan and Andrew Zisserman at the University of Oxford. It is a popular object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy and is easy to use with transfer learning.

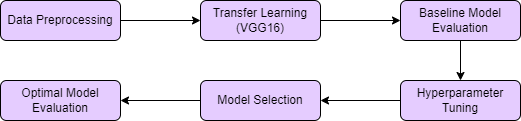
The architecture of VGG16 is as follows.



*Figure 1: VGG16 Architecture*

# **3 Methodologies**

## **3.1 Overview Of Approach**



*Figure 2: Overview Of Classification Approach*

# **3.2 Data Preprocessing**

We will be using two datasets in this project, namely CelebA and Adience.

### *3.2.1 CelebA*

Celeba is a larger data set with images that are mostly frontal and well lit, and is not representative of images in the wild. The intention is to perform pretraining and hypertunning of the models on this dataset to reduce possibility of any anomalous results in the evaluation of the model due to the data set. Note that:

* We will be ensuring the ratio of the training data is 1:1 ratio, ie 5000 male and 5000 female. This will help reduce representation bias
* We will also remove all the nan values in the relevant columns for the datasets

### *3.2.2 Adience*

Adience is a dataset obtained from Flickr and hence is representative of in the wild images. We will be using this data set for final evaluation of the model, as well as comparing model degradation when the data set is moved from a “Sterile dataset” to “Dirty Dataset”. Note that:

* The Adience dataset is split into 5 fold, with each fold being independent from the rest. This allows us to easily perform k fold cross validation.
* We will being using a 60/20/20 train/validation/test split for Adience.
* We will also remove all the nan values in the relevant columns for the datasets

*3.2.3 Preprocessing*

Both data sets would be normalised and resized to 224x224 prior to being passed to the model.

## **3.3 Evaluation Metrics**

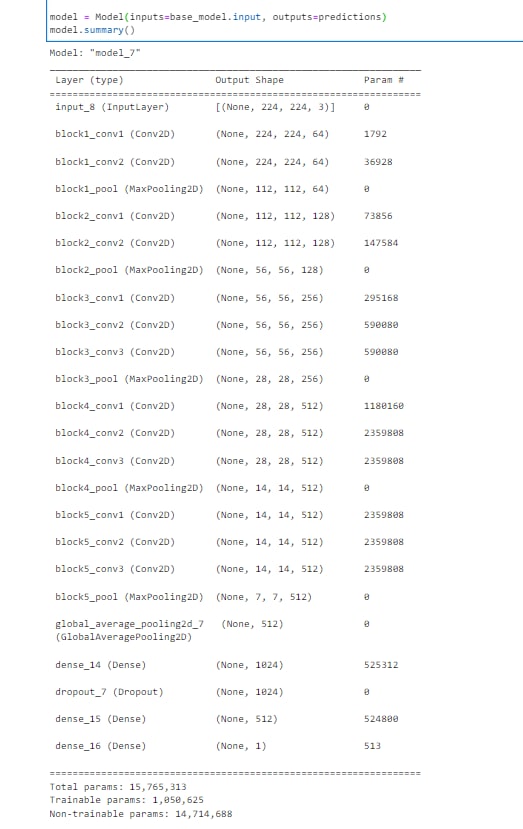
We utilise both validation accuracy and validation loss to evaluate the performance of our model during training. We implement early stopping to prevent our model from overfitting. We set the validation loss as the metric to be monitored with a patience to be 10. This allows us to ensure that the training terminates at slightly past the global minima, and the training does not stop at a particularly steep local minimum.

The final performance of the model is determined by test accuracy and test loss.

## **3.4 Transfer Learning with VGG16**

We first obtained a baseline model using VGG16 backbone, with a Global average pooling layer, 2 fully connected and 2 dropout layers. The output is then fed into a sigmoidal out layer which performs the classification task.

The configuration of our model is as follows.



*Figure 3: Baseline VGG16-CNN Architecture*

The VGG16 base is implemented using the tensorflow.keras.applications VGG16 module, with the weights of the base model being instantiated to the ones pre-trained on ImageNet. The VGG16 models with these weights are then frozen. This is followed by a global average max pooling layer. The output of these layers is then fed to the 2 fully-connected layers,both with Relu activation functions and 1024 and 512 neurons respectively. Between the fully-connected layers is a dropout layer with a rate of 0.5. Finally, this is passed to an output sigmoidal layer, which produces a classification label. Our model also makes use of the Adam optimiser, has a learning rate of 0.0001, and a batch size of 16. Adam was used for models trained on CelebA as they would be further trained on Adience. Hence a quicker convergence was preferred. The models would then be further fine tuned on Adience with a more conservative optimiser(SGD w/ Momentum) to obtain a better result.

### *3.4.1 Baseline Model Evaluation*

We then train our modified CNN model on the CelebA dataset, which is a large-scale face attributes dataset with more than 200k images. As CelebA is a huge dataset, we chose to consider only a 14k subset of it. Out of these, 10k will be used as the train set, 2k as the validation set and the remaining 2k as the test set.

The performance of our baseline VGG16-CNN model on CelebA are as follows.

|  | **Accuracy** | **Loss** |
| --- | --- | --- |
| Validation | 0.936    *Figure 4: Validation Accuracy Of Baseline Model (CelebA)* | 0.170    *Figure 5: Validation Loss Of Baseline Model (CelebA)* |
| Test | 0.908 | 0.246 |

*Table 1: Baseline Model Performance (CelebA)*

## **3.5 Hyperparameters Tuning**

We utilised the Keras Hyperband Tuner to conduct our hyperparameter tuning.

### *3.5.1 Hyperparameter Choice*

* VGG16 Layers
* Number of frozen layers

Rationale: In the baseline model, all of the VGG16 layers were frozen, to preserve feature extraction capabilities that were developed in the imagenet training. However, as part of hyperparameter tuning, we opted to unfreeze portions of the VGG16 backbone to determine if this would improve performance. This is as the feature extraction in the later layers of the original network might have been overly generalised for the purposes of our task as it was initially used on a dataset of 1000 classes. The features in a binary classification problem might achieve better results with a different set of weights. Hence we added a number of layers to unfreeze as an additional hyperparameter to tune.

* Customised Layers
* Number of neurons in fully-connected layers
* Dropout rate in dropout layers
* Learning rate of Adam optimizer

#### *3.5.2 Tuning Approach*

In the interest of saving computational time, we performed two rounds of hyperparameter tuning, the first instance was with a larger search range and step, to locate to approximate location more quickly. Before performing the second tuning with a smaller range and greater granularity to find the most optimal hyper parameters.

We initially performed tuning with a larger step, with the following hyperparameters. The second tuning was performed with a smaller range and greater granularity to find the most optimal hyper parameters.

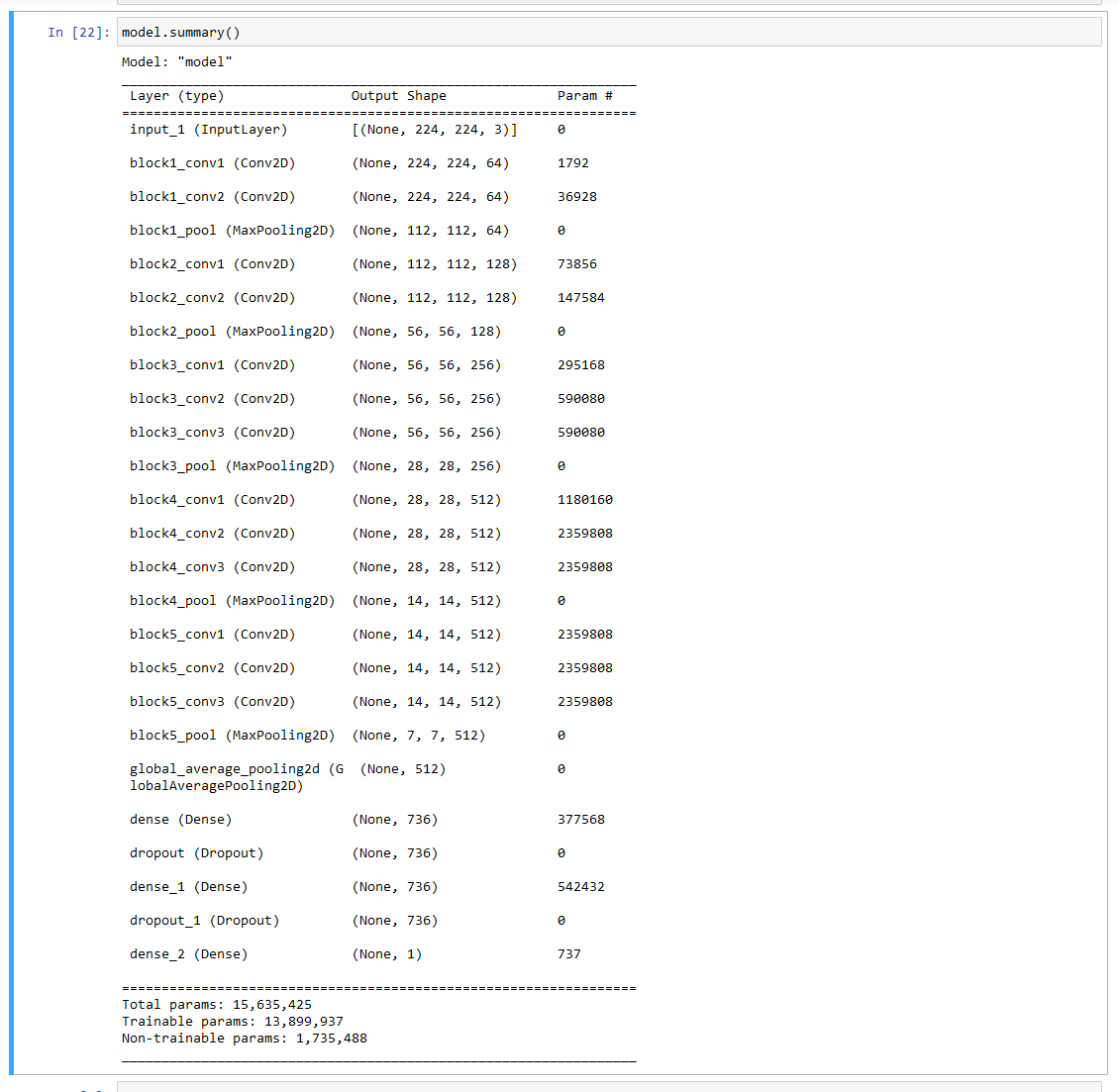
The hyperband turning the produced the following results:

| **Hyperparameter To Tune** | **1st Round Tuner Values** | **2nd Round Tuner Values** | **Optimal Values Found** |
| --- | --- | --- | --- |
| No. of neurons in 1st dense layer | [32:1024:32] | [688:784:16] | 736 |
| No. of neurons in 2nd dense layer | 752 |
| Dropout rate of 1st dropout layer | [0.1:0.5:0.1] | [0.1, 0.5, 0.8] | 0.1 |
| Dropout rate of 2st dropout layer | 0.5 |
| Learning rate of optimizer (Adam) | 0.01, 0.001, 0.0001 | - | 0.0001 |
| No. of frozen layers | [8:16:2] | [8:12:1] | 10 |

*Table 2: Hyperparameter Tuning Experiments and Results*

## **3.6 Model Selection**

The configuration of our optimal model is as follows.

0.

*Figure 6: Optimal VGG16-CNN Architecture*

### **3.6.1 Optimal Model Evaluation**

The performance of our optimal VGG16-CNN model on CelebA are as follows.

|  | **Accuracy** | **Loss** |
| --- | --- | --- |
| Validation | 0.975  *Figure 7: Validation Accuracy Of Optimal Model (CelebA)* | 0.083    *Figure 8: Validation Loss Of Optimal Model (CelebA)* |
| Test | 0.963 | 0.109 |

*Table 3: Optimal Model Performance (CelebA)*

# **4 Novelties**

## **4.1 Pre-Training on CelebA and Transfer Learning with Adience**

The facial images of the CelebA dataset are quite clean and controlled, in a sense that most of them are frontal facing and taken in good lighting and resolution. As such, we want to see how our optimal model will perform on more dirty images, which are more reflective of real-world use cases. The Adience dataset has more than 25,000 images from Flickr, that attempt to capture all the variations in appearance, noise, pose, lighting and more, intending to be as true to the challenges of real-world imaging conditions. We thus perform transfer learning by applying the best model we obtained from training on the CelebA dataset, to the Adience dataset. In addition, instead of randomly initialising the weights for training on the Adience dataset, we will be using the saved weights obtained from training our model on CelebA. This concept is known as pre-training, which reduces training time and resources.

### *4.1.1 Results Analysis*

The performance of our optimal VGG16-CNN model on Adience are as follows.

|  | **Accuracy** | **Loss** |
| --- | --- | --- |
| Validation | 0.912    *Figure 9: Validation Accuracy Of Optimal Model (Adience)* | 0.241    *Figure 10: Validation Loss Of Optimal Model (Adience)* |
| Test | 0.854 | 0.346 |

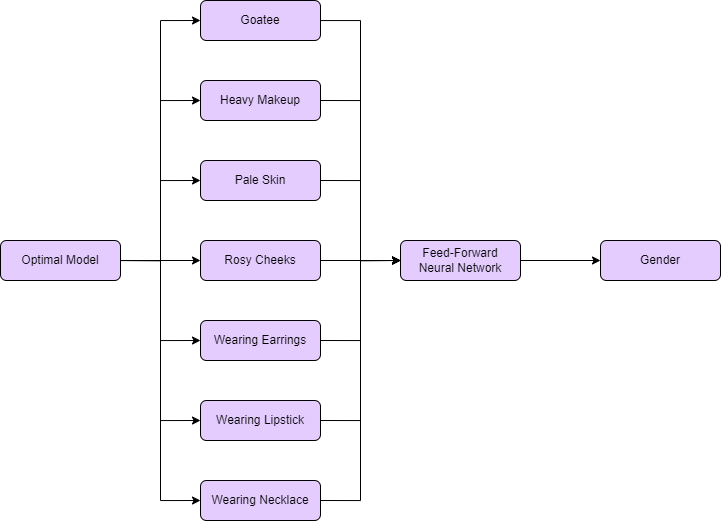
*Table 4: Optimal Model Performance (Adience)*

## **4.2 Feature Importance**

### *4.2.1 Rationale*

When classifying whether someone is male or female, there are many facial features to consider. The goal here is to find out what facial features are the most important to determine whether an image of a person should be classified as male or female. This is with the intent of when a more ambiguous image of a person is received by the model, instead of solely relying on the image to make a prediction, the other predicted features can also play a part in aiding the model’s decision-making process.

### *4.3.2 Approach*



*Figure 11: Overview Of Feature Importance Approach*

In addition to gender, the images in the CelebA dataset have many other labels to them, describing different facial features. We hence extract 7 of these features to apply our optimal model on. These 7 predicted outputs will then be fed into another feed-forward neural network as binary feature inputs, with gender as the corresponding output. We then find the importance of each of these features.

The following are the list of 7 features we have chosen for consideration.

* Goatee
* Heavy Makeup
* Pale Skin
* Rosy Cheeks
* Wearing Earrings
* Wearing Lipstick
* Wearing Necklace

### *4.3.3 Evaluation*

#### *4.3.3.1 Evaluation Metrics*

Test accuracy will be used to evaluate both the performance of our optimal model in predicting each feature while both the test accuracy and loss will be used to predict the performance of our newly defined feed-forward neural network using these features to predict gender.

The importance of each feature is evaluated by calculating their SHapley Additive exPlanations (SHAP) values and displaying them in a Force Plot. SHAP Values break down a prediction to show the impact of each feature. They do so by interpreting the impact of having a certain value for a given feature in comparison to the prediction we would have made if that feature took some baseline value.

#### *4.3.3.2 Evaluation of Optimal Model in Predicting Each Feature*

| **Feature** | **Test Accuracy** |
| --- | --- |
| Goatee | 0.814 |
| Heavy Makeup | 0.721 |
| Pale Skin | 0.825 |
| Rosy Cheeks | 0.828 |
| Wearing Earrings | 0.801 |
| Wearing Lipstick | 0.629 |
| Wearing Necklace | 0.827 |

*Table 5: Optimal Model Performance On Other Features*

#### *4.3.3.3 Evaluation of Feed-Forward Model in Predicting Gender*

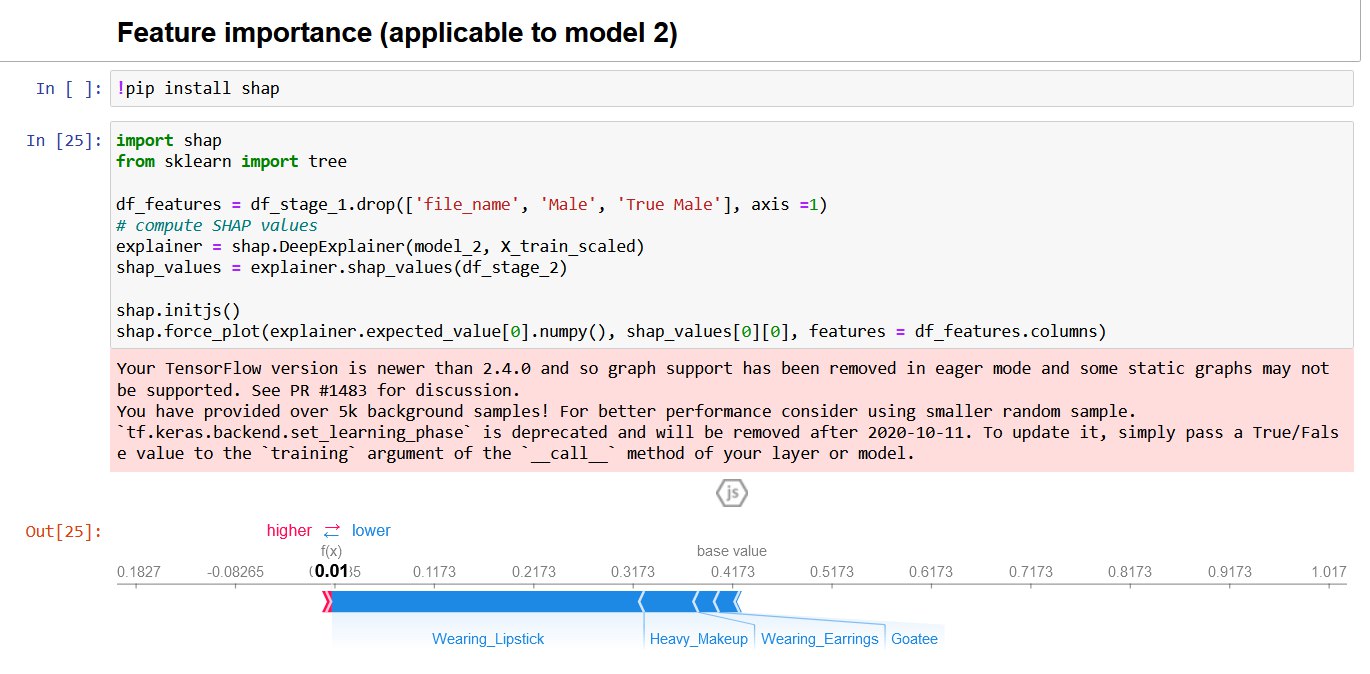
| **Test Accuracy** | **Test Loss** |
| --- | --- |
| 0.912 | 0.234 |

*Table 6: Feed-Forward Neural Network Performance On Gender*

#### *4.3.3.4 Evaluation of Feature Importance*

| **Rank** | **Feature** |
| --- | --- |
| 1st | Wearing Lipstick |
| 2nd | Heavy Makeup |
| 3rd | Wearing Earrings |

*Table 7:Top 3 Most Important Features For Gender Prediction*



*Figure 12: Force Plot Of SHAP Values*

From the Force Plot, it can be observed that the features that push the prediction value closer to 0 (female) are significantly more important than those pushing the prediction value to 1 (male). The blue factors all take up a much larger area than the red factors. Notably, the top 3 factors all belong to the blue area that pushes the value to 0, explaining why the value observed is as close to 0 as it is, i.e. 0.01.

# **5 Discussion**

## **5.1 Discussion On Optimal Model Performance On CelebA Versus Adience**

From the results of our project, it is noted that our model performs about 10% better when applied to CelebA images as compared to when applied to Adience images. This is expected as we train our model using CelebA images and it thus better adapts to the context of those images. However, we note that our model still performs quite well on the Adience images, suggesting it is quite a good solution for us to mitigate the challenges mentioned in our problem statement. A possible suggestion to make our model perform better on Adience is conducting further hyperparameter tuning to the model that is ported over. The results of our hyperparameter tuning of the model on CelebA, can be used as a good starting point to narrow down the range of our hyperparameters to re-tune. Another suggestion to improve model performance is running a few concurrent models on the same image and taking a majority vote on the predicted labels, in order to decide on one. For instance, if 4 out of the 5 models we passed the image through predicted that the person in the image is female while only 1 out of the 5 models predicted it as male, we will assign the label of this image to be female.

## **5.2 Discussion On Hyperparameters Tuning Experiments**

In our project, we note that hyperparameter tuning is not an easy task as there is no way of knowing what range of values to consider. Specifying too many hyperparameters with too huge ranges will both be computationally costly, and with unrealistic training times. Some tips we have picked up on while conducting hyperparameter tuning on our model is that more than one round of tuning can be conducted. In each round, we can specify different hyperparameters to tune each time. For instance, we can tune the batch size and learning rate in the first tuner, which helps reduce training time and optimal parameters are searched more quickly in subsequent tuners. We can also narrow down the range of values to be tuned per hyperparameter at each round of tuning, which is what we tried out for our project. In addition, not only the conventional hyperparameters can be considered for tuning. We can define our own such as the number of layers to freeze in our backbone model (VGG16).

## **5.3 Discussion On Selection of Features For Feature Importance**

The CelebA dataset has 40 different labels for each image. In our project, we only consider a subset of 7 of these features to conduct experiments on. In particular, we chose features that are highly gender specific, in order to obtain ideal results that clearly display our intentions. In other words, one intuitively knows that if the person in the image has a goatee, there is a very high chance for it to be classified as a male. On the other hand, if the person in the image is wearing lipstick, it will likely be classified as a female. However, in reality, there are some features that are not so clear cut such as whether the target image has brown hair or not. If we consider all of the given labels, our feature importance results are likely to be very different and have more ambiguity.

# **6 References**

[1] Brownlee, J. (2020, Aug 25). *Use Early Stopping to Halt the Training of Neural Networks At the Right Time*. Machine Learning Mastery. Retrieved November 11, 2022, from https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/

[2] Brownlee, J. (2022, Aug 7). *Evaluate the Performance of Deep Learning Models in Keras*. Machine Learning Mastery. Retrieved November 11, 2022, from https://machinelearningmastery.com/evaluate-performance-deep-learning-models-keras/

[3] Dewasme, L. (2018, November 4). *Gender Classification Based on Multiscale Facial Fusion Feature*. Hindawi. Retrieved November 11, 2022, from https://www.hindawi.com/journals/mpe/2018/1924151/

[4] Gül, R. (n.d.). *Age & Gender Detection: Top Use Cases*. Cameralyze. Retrieved November 11, 2022, from https://www.cameralyze.co/blog/age-gender-detection-top-use-cases

[5] Hassan, M. u. (2018, November 20). *VGG16 - Convolutional Network for Classification and Detection*. Neurohive. Retrieved November 11, 2022, from https://neurohive.io/en/popular-networks/vgg16/

[6] Thite, A. (2021, Oct 5). *What is VGG16 - Convolutional Network for Classification and Detection*. Great Learning. Retrieved November 11, 2022, from https://www.mygreatlearning.com/blog/introduction-to-vgg16/

[7] Trevisan, V. (2022, Jan 18). *Using SHAP Values to Explain How Your Machine Learning Model Works*. Towards Data Science. Retrieved November 11, 2022, from https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137