NATIONAL UNIVERSITY OF SINGAPORE

Faculty of Science

AY 2021/2022, SEMESTER 2

DSA4212: Optimisation for Large-Scale Data-Driven Inference

Assignment 1

Done by: Group 12

Chua Xin Xuan (A0205767X)

Joey Tan Xin Yi (A0206334N)

Quek Su Ning (A0205557A)

Tan Jie Yi (A0206383H)

1 Project Description

In this project, we aim to build variants of a logistic regression model to predict if an image corresponds to a Male or a Female. In particular, we have built a basic Logistic Regression Model, Ridge Logistic Regression Model and a Lasso Logistic Regression model. The test accuracy was compared among the various optimization algorithms such as Naive Gradient Descent, Automatic Step Size selection and Stochastic Gradient Descent.

2 Exploratory Data Analysis

The data consists of 20000 images of male and female faces. The first 15000 images are used for training and the last 5000 images are used for testing.

There are 8431 males and 11569 females in the training data. The ratio of males to females is 4:6 which is relatively balanced.

The faces of an average male and female are shown below in *Figure 1*:

Figure 1

3 Models

For logistic regression, the MLE β_* is the solution of the optimization problem

$$\beta_* = \mathbf{argmin} \left\{ \beta \mapsto \sum_{i=1}^{N} \mathbf{Loss}(\beta, x_i, y_i) \right\}$$

where $Loss(\beta, x_i, y_i) = log \left\{ 1 + e^{-y_i \langle \beta, x_i \rangle} \right\}$

Unlike linear regression, there is no closed-form solution for logistic regression. However, since the loss function is convex, we can explore the different optimization techniques to minimize the loss function.

We will be exploring the following optimization algorithm: Naive Gradient Descent with Fixed Step Size, Naive Gradient Descent with Automatic Step Size Selection and Stochastic Gradient Descent

These models will act as baseline models and we seek to create a model that has a higher accuracy.

For all models, we set the random seed as 1, to initialize beta for a more accurate comparison of the different models. This also ensures that the results are reproducible.

3.1 Naive Gradient Descent with Fixed Step Size

Using a fixed step size of 5×10^{-7}, we applied Naive Gradient Descent to optimize our basic logistic regression model. Since logistic regression is a convex function, Naive Gradient Descent is able to optimize the function and allow the function to converge to a global minimum. The loss history plot and ROC Curve with AUC is as shown in *Figure 2*.

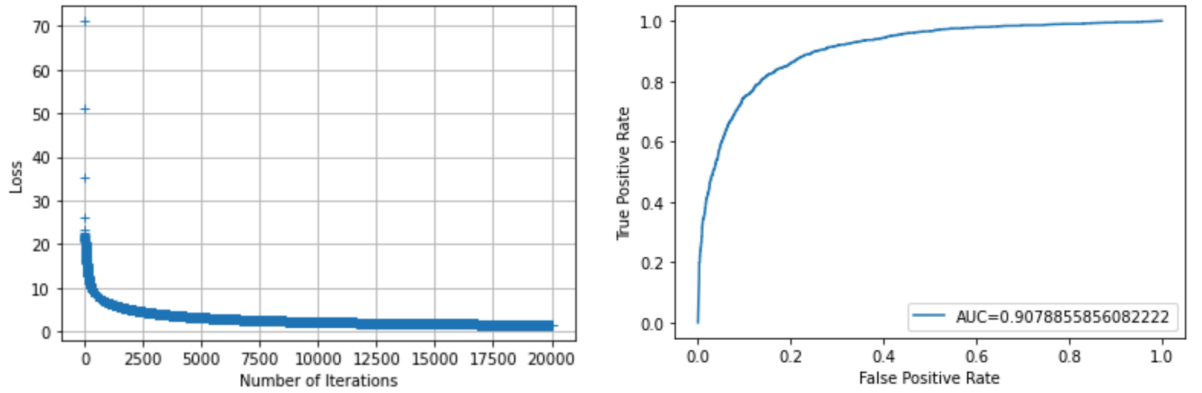

Figure 2

Training Accuracy	84.5%
Testing Accuracy	83.6%
Area Under Curve	0.908

As shown in *Figure 2*, Naive Gradient Descent allows the logistic regression model to converge slowly to a minimum. With 20,000 iterations, training and testing accuracy reached 84.5% and 83.6% respectively. It is likely that Naive Gradient Descent can reach a lower loss value and achieve higher accuracies with more iterations. However, due to the relatively long time that Naive Gradient Descent takes to converge, we explore other optimization algorithms to evaluate different preprocessing techniques.

3.2 Automatic Step Size Selection

Using an initial $\eta = 1$ and $\alpha = 0.5$, We used backtracking to find an optimal step size $\eta_n > 0$ such that $F(x_n + \eta_n d_n) \leq F(x_n) + \alpha \eta_n \langle \nabla F(x_n), d_n \rangle < F(x_n)$, where d_n is the decent direction. This is done over 20000 iterations as well. The loss history and ROC curve can be seen in *Figure 3* below:

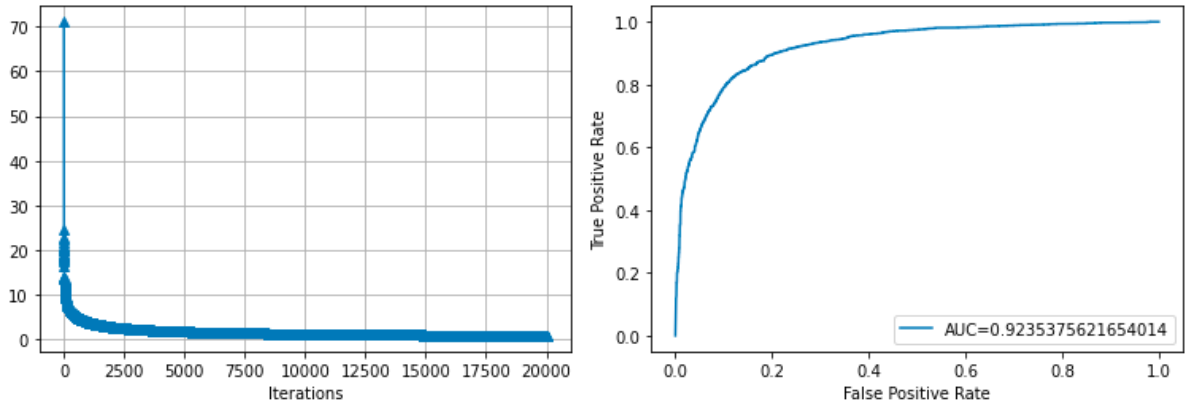

Figure 3

Training Accuracy	88.3%
Testing Accuracy	85.6%
Area Under Curve	0.924

With 20000 iterations, the training and testing accuracy reached 88.3% and 85.6% respectively.

3.3 Stochastic Gradient Descent

Stochastic Gradient Descent works by replacing the actual gradient by an estimation of it. It works well in high dimension problems as it reduces the computation burden of calculating the actual gradient. An estimation of the gradient is given by

$$\widehat{\nabla F}(x) = \frac{1}{n}\sum_{k=1}^{n} \nabla_x L(x, y_{ik})$$

A learning rate of 5×10^{-9} and minibatch size of 5 was used. The model ran over 1500 epochs. The loss history plot and ROC Curve with AUC is as shown in *Figure 4*.

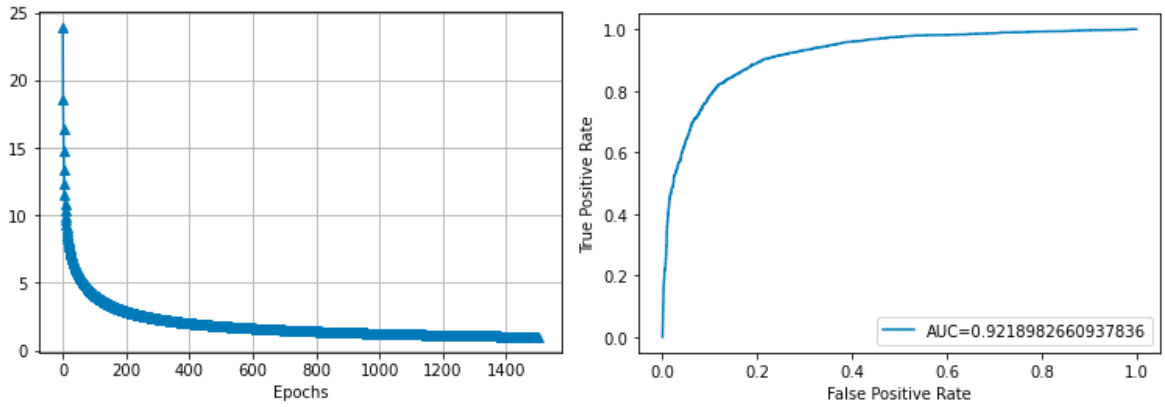

Figure 4

Training Accuracy	87.6%
Testing Accuracy	85.4%
Area Under Curve	0.922

With 1500 epochs, the training and testing accuracy reached 87.6% and 85.4% respectively.

To conclude, while the test accuracy of the Naive Gradient Descent with automatic step size algorithm is slightly higher than when Stochastic Gradient Descent is used, we note that it took the former model much more iterations to reach a comparable accuracy as the latter. The time taken to run 20000 iterations was 8984 seconds as compared to 1204 seconds for the 1500 epochs.

In this section, we have explored the different optimization algorithms. These models will act as baseline models and we seek to create a model that has a higher accuracy.

4 Specific Tasks

In our further analysis, we chose to use the Stochastic Gradient Descent optimization algorithm with fixed learning rate as it has achieved a high accuracy in a faster time as compared to the other two optimization algorithms on our original grayscale dataset. In addition, since we are dealing with high dimensional data, we should take the computational cost into account when exploring the effect of different image preprocessing techniques. We note that Stochastic Gradient Descent converges faster as it updates the parameters more frequently.

However, we would be using both Stochastic Gradient Descent and Naive Gradient Descent with backtracking algorithms on RGB images to investigate if Stochastic Gradient Descent will do better than Naive Gradient Descent on RGB images.

4.1 Impact of Size of Training Set on Test Accuracy (Task 1)

4.1.1 Grayscale Images

We investigated the effect of different training set sizes on the test accuracy of the last 5000 images. For this analysis, we used Stochastic Gradient Descent as our optimization algorithm on our basic logistic regression model for various training set sizes, ranging from 1000 to 15000. A learning rate of 5×10^{-9} and mini-batch size of 5 was used for all training sizes. The test accuracy over 1500 epochs was plotted for each training set size as shown in *Figure 5*.

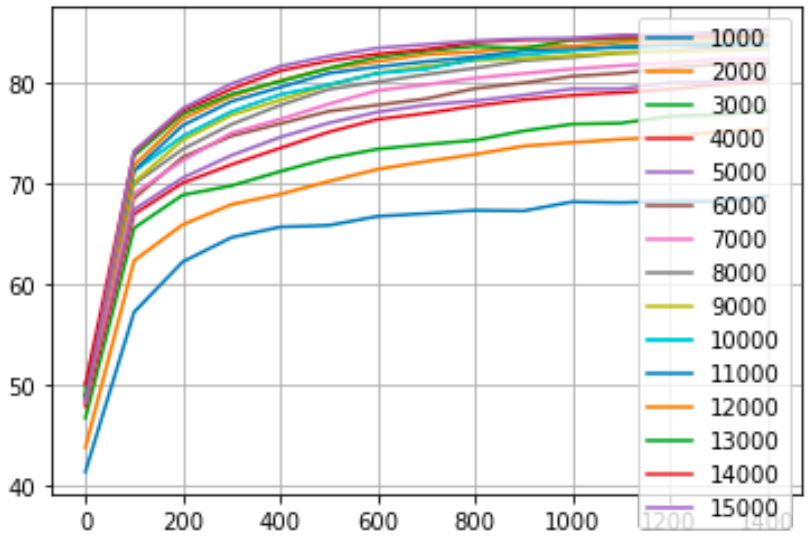

Figure 5

It can be observed that the test accuracy curves are clustered together for training sets with a size of about 7000 and above. This implies that not all 15000 images are needed to train the model as 7000 images could be a large enough dataset.

4.1.2 RGB Images

The same procedure was also performed on RGB images to see if less images were required before the accuracy stabilizes. The test accuracy over 1500 epochs was plotted for each training set size as shown in *Figure 6*.

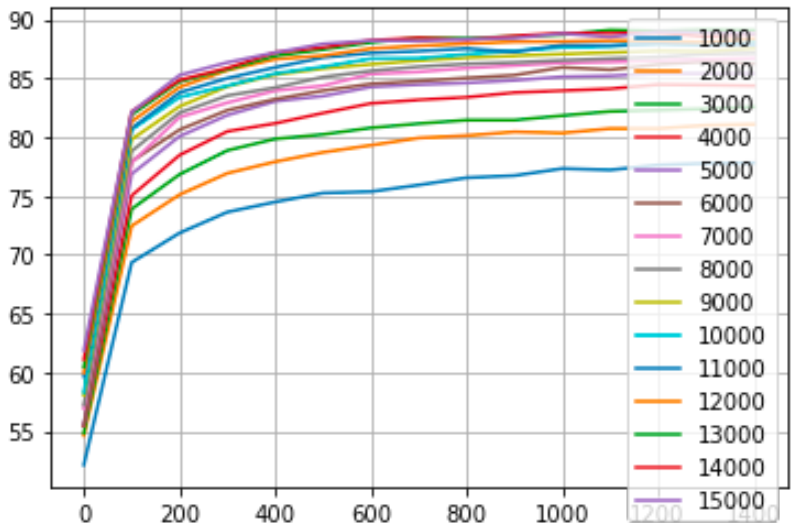

Figure 6

It can be observed that the test accuracy curves are clustered together for training sets with a size of about 5000 and above. Moreover, we note that the test accuracy achieved is higher than that of grayscale images where 3000 RGB images were already able to achieve the test accuracy of using 10000 grayscale images. This will be further discussed in Section 4.3 on the Impact of Color on Test Accuracy.

4.2 Impact of Resolution of Images on Test Accuracy (Task 2)

Image resolution describes how many pixels are displayed per inch of an image. Images with higher resolution have more pixels displayed per inch, hence resulting in better quality than images with lower resolution. Originally, the Celeba images have a resolution of 218 by 178. We converted the RGB images in the dataset to grayscale before changing the resolution. To investigate the impact of image resolution on test accuracy, we modified the images in the data set to take on different resolutions, namely 200 by 200, 150 by 150, 100 by 100, 50 by 50, and 20 by 20. The leftmost image in *Figure 7* is the original image in the dataset while the 5 subsequent images show the same image in different resolutions.

Figure 7

Subsequently, we ran Stochastic Gradient Descent on our basic logistic regression model for each of the 15000 training images in different resolutions and tested the accuracy of the models on the test set. We ran the model over 1500 epochs and mini-batch size = 5 and used a fixed learning rate of 5×10^{-9}.

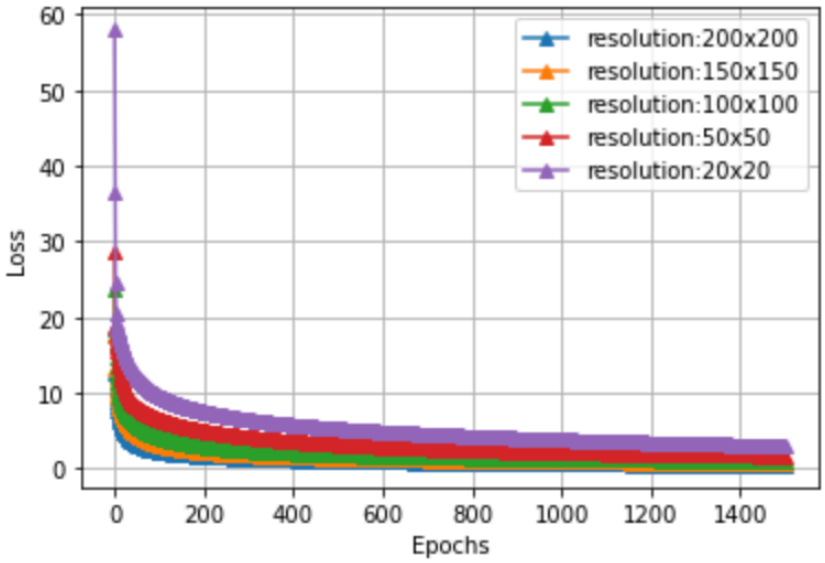

Figure 8

Resolution	20 x 20	50 x 50	100 x 100	150 x 150	200 x 200
Training Accuracy	66.3%	80.7%	87.7%	90.3%	92.3%
Testing Accuracy	65.2%	79.6%	85.5%	85.6%	86.1%

Table 1

As shown in *Figure 8* and *Table 1*, the dataset consisting of images with the highest resolution 200 by 200 has the lowest loss and highest training and testing accuracy, followed by the next highest resolution and so on. This trend is expected since more information is retained when a higher resolution is used.

However, we note that using images with slightly lower resolutions of 150 by 150 or 100 by 100 gives a comparable result to using images with a higher resolution of 200 by 200. Hence, it may be sufficient to train the model with images with a 150 by 150 or 100 by 100 resolution.

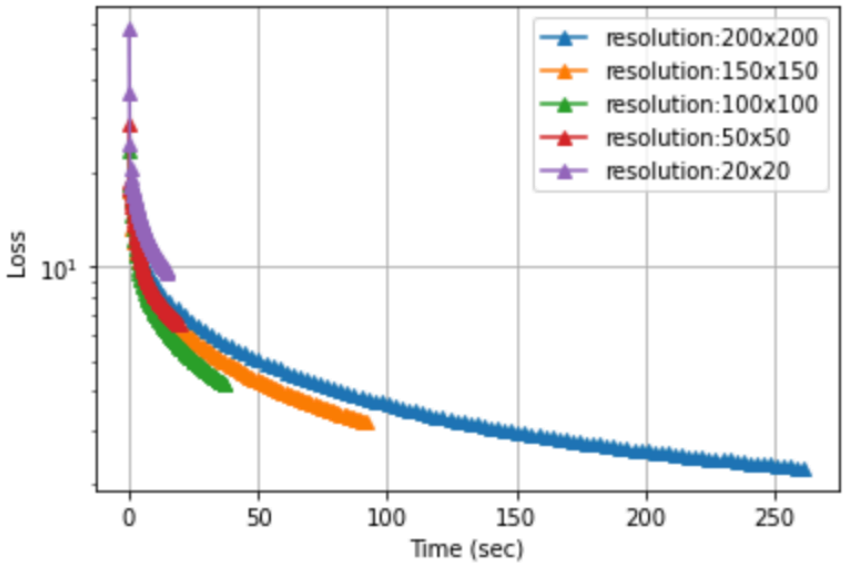

Figure 9

In addition, *Figure 9* illustrates that when using only 100 epochs, different image resolutions have already resulted in a significant difference in the time needed to train the model. The logistic regression model takes a longer time to train when the training dataset consists of images with higher resolution. Taking time complexity into account, we consider using images with a slightly lower resolution of 100 by 100 for further exploration of other preprocessing techniques.

4.3 Impact of Coloured Image (Task 3)

4.3.1 Test Accuracy of Coloured Image vs Grayscale image

We will be exploring the effects of RGB images vs Grayscale images on both Automatic Step Size selection model and Stochastic Gradient Descent model.

4.3.1.1 Automatic Step Size Selection model:

For the Automatic Step Size Selection model, similar to the method used previously, we start with a learning rate of 1 and use backtracking with $\alpha = 0.5$. To have a fairer comparison between the model that is trained on grayscale images and the model that is trained on RGB images, the accuracies of both models were compared after 300 iterations.

We achieved the following loss history plot and ROC Curve as seen below in *Figure 10*:

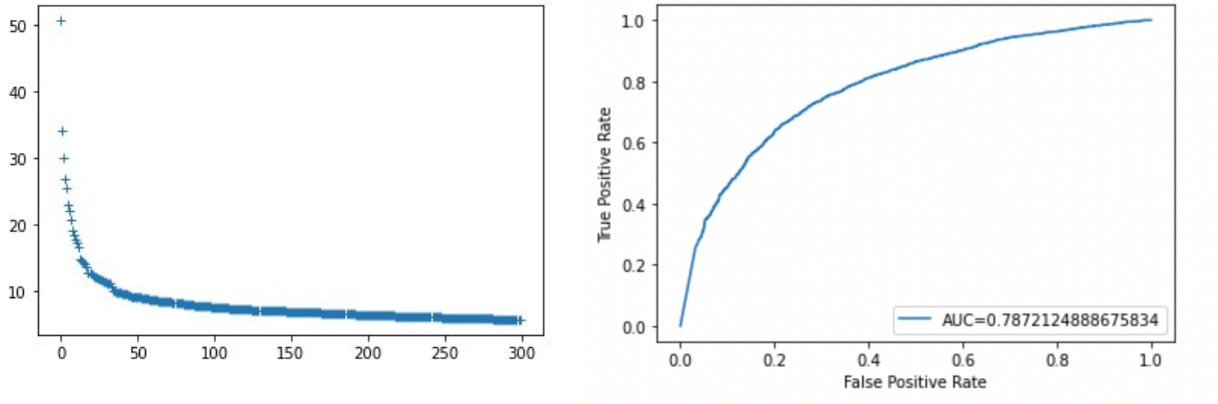

Figure 10

	Grayscale Images	**RGB Images**
Training Accuracy	67.0%	72.6%
Testing Accuracy	68.2%	73.0%

Table 2

From *Table 2*, we observed that the model which ran on RGB images did significantly better than Grayscale Images on both training and testing data for both models. The AUC value was 0.787.

4.3.1.2 Stochastic Gradient Descent model:

For this Stochastic Gradient Descent model, we ran on 1000 epochs, mini-batch size of 5 and learning rate of 5×10^{-9}, and achieved the following loss history at the end of each epoch plot and ROC Curve shown in Figure 11:

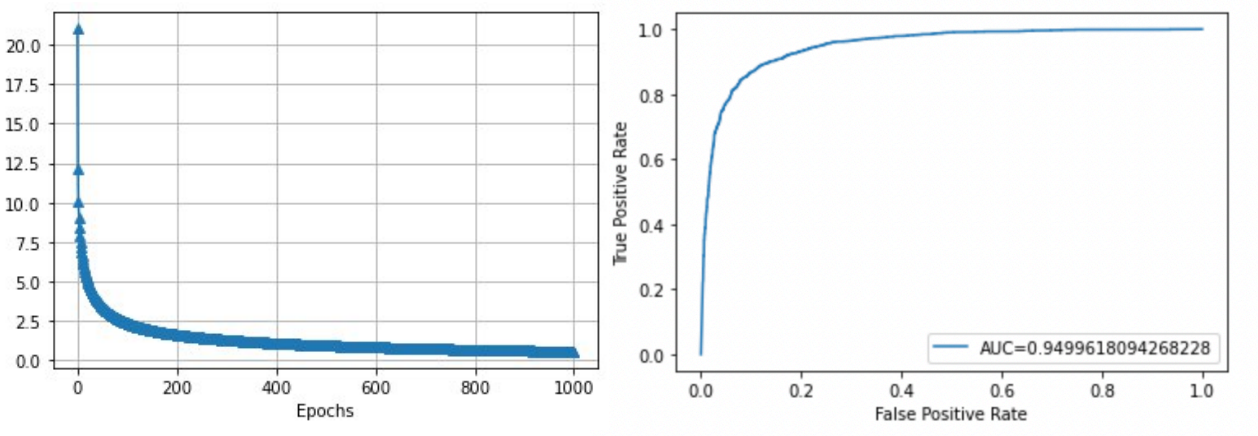

Figure 11

	Grayscale Images	RGB Images
Training Accuracy	87.6%	92.6%
Testing Accuracy	85.4%	88.7%

Table 3

Similarly, from *Table 3*, we observed that the model which ran on RGB images did better than Grayscale Images on both training and testing data. The AUC value was 0.950.

In conclusion, coloured images are a significant feature which help to distinguish between Males and Females. Using RGB images to train the model has resulted in a higher train and test accuracy.

However, we also note that using RGB images results in a much higher computational cost. Also, it requires a higher memory as coloured images have 3 channels, while grayscale images only have 1 channel.

Furthermore, Stochastic Gradient Descent also significantly outperforms Automatic Gradient Descent on both training and test accuracy as seen above.

4.3.2 Impact of Image Pre-Processing

We will be experimenting with the contrast, brightness and saturation of the images using the Python Imaging Library (PIL) to observe its impacts on test accuracy.

For cross-validation, we split the training set of the first 15,000 images randomly into 70% for training and 30% for validation. We used Gradient Descent with Automatic Step Size Selection on varying intensity of contrast, brightness and saturation to observe the effects it has on the test accuracy respectively. The model ran on 300 iterations, with a learning rate of 1 and backtracking with $\alpha = 0.5$.

4.3.2.1 Contrast

Figure 12

From *Figure 12*, we observe the images with varying contrast intensity between 0.7 and 2.0 at intervals of 0.1. Images with contrast intensity less than 1.0 have a lower contrast, while images with contrast intensity greater than 1.0 have higher contrast. The original image is observed when the contrast intensity is 1.0.

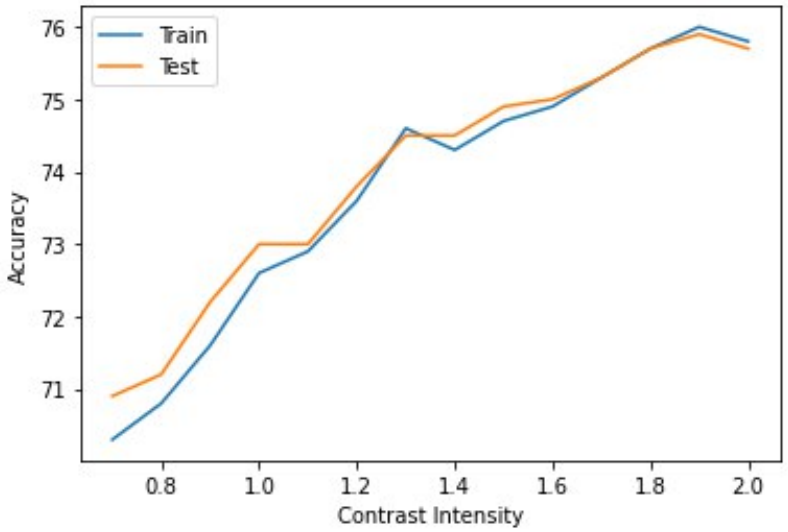

Figure 13

From *Figure 13*, when contrast intensity increases, there is a general increase in both training and testing accuracy. Test accuracy peaked when contrast intensity was 1.9, with training accuracy of 76.0% and testing accuracy of 75.9%. As compared to the original intensity of the images, which has a lower training accuracy of 72.6% and testing accuracy of 73.0%.

Hence, we conclude that an increase in contrast improves test accuracy. However, any contrast intensity higher than 2.0 is likely to result in poorer fit, as high contrast might lead to loss of important features in the images. On the other hand, a decrease in contrast will lead to lower test accuracy.

4.3.2.2 Brightness

Figure 14

From *Figure 14*, we observe the images with varying brightness intensity between 0.7 and 1.5 at intervals of 0.1. Images with brightness intensity less than 1.0 had a lower brightness, while images with brightness intensity greater than 1.0 had higher brightness. The original image is observed when the brightness intensity is 1.0.

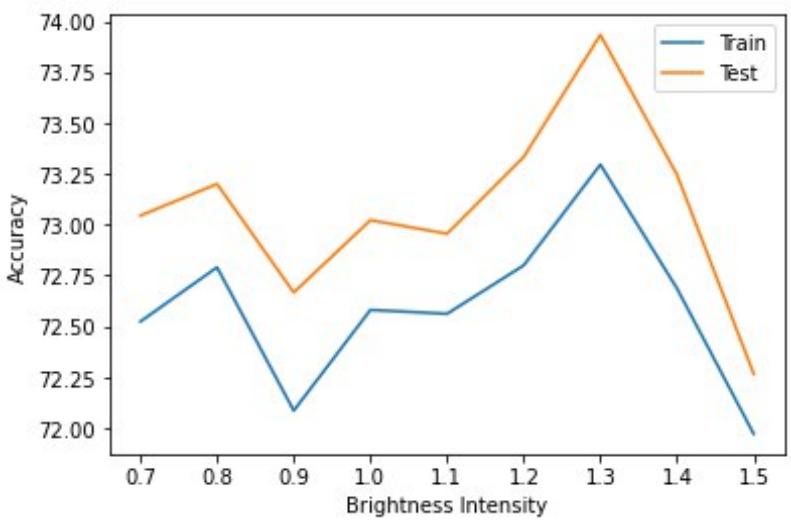

Figure 15

From *Figure 15*, we observe that increasing brightness intensity would improve training and testing accuracy. However, at extreme brightness intensity, the accuracy drops. This is likely because high brightness intensity would lead to overexposed images, which results in loss of important features in the images.

Accuracy peaked when brightness intensity is at 1.3, with training accuracy of 73.3% and testing accuracy of 73.9%. As compared to the original intensity of the images, which has a lower training accuracy of 72.6% and testing accuracy of 73.0%. However, accuracy drops drastically when brightness intensity is greater than 1.3.

Since there is a small improvement of 0.9% in test accuracy only, increasing brightness is less significant than increasing contrast, which has seen a greater improvement of 2.9% in test accuracy.

4.3.2.3 Saturation

Figure 16

From *Figure 16*, we observe the images with varying saturation intensity between 0.6 and 2.8 at intervals of 0.2. Images with saturation intensity less than 1.0 has a lower saturation, while images with saturation intensity greater than 1.0 had higher saturation. The original image is observed when the saturation intensity is 1.0.

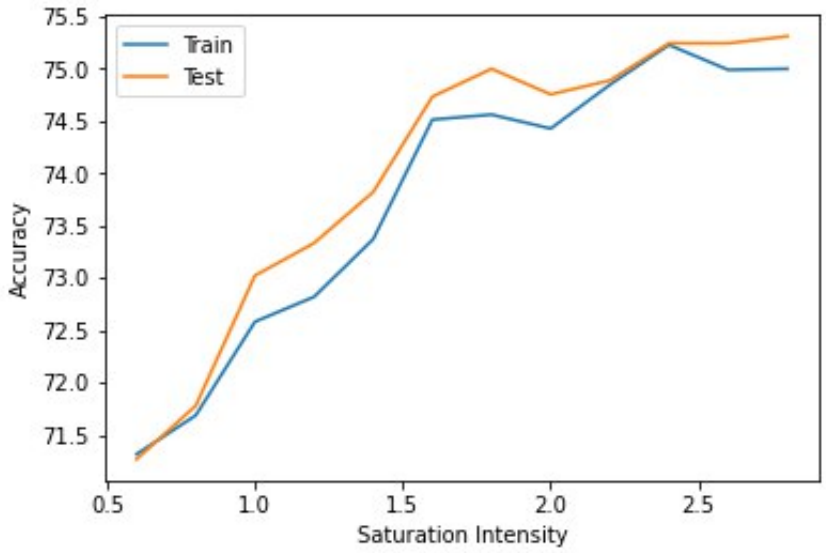

Figure 17

From *Figure 17*, when saturation intensity increases, there is a general increase in both training and testing accuracy. Test accuracy peaked when saturation intensity was 2.8, with training accuracy of 75.0% and testing accuracy of 75.3%. As compared to the original intensity of the images, which has a lower training accuracy of 72.6% and testing accuracy of 73.0%.

Hence, we conclude that an increase in saturation improves test accuracy.

4.4 Using Specific Features (Task 4)

Logistic regression models were trained using only specific features of the face such as front face and eyes. Different learning rates and different number of epochs may be used depending on the dataset changes. We noted that some learning rates may work on certain parts of the face but some gave nan values for the loss function or did not converge.

4.4.1 Eyes

We used a pretrained classifier “haarcascade_eye.xml” to detect the location of eyes and ran a Stochastic Gradient Descent model on 21777 RGB training images of the eyes. The test accuracy was computed on 7244 RGB test images of the eyes. The model ran over 1500 epochs, and a mini-batch size of 5. The learning rate is 5×10^{-7}. The loss history at the end of each epoch and ROC Curve are shown in the plots below in *Figure 18*:

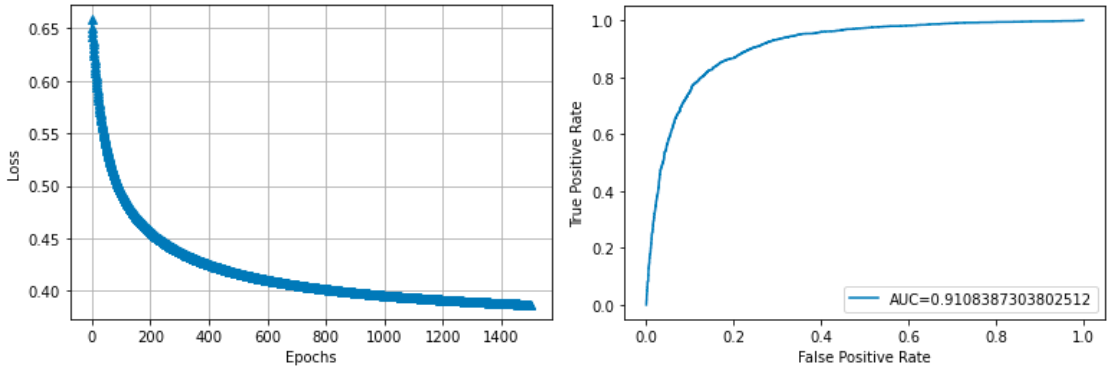

Figure 18

	RGB Images
Training Accuracy	83.9%
Testing Accuracy	84.7%

The test accuracy obtained using RGB images was 84.7%. Its testing accuracy is slightly lower as compared to the test accuracy of the model on the full face, which has a testing accuracy of 88.7%.

4.4.2 Nose

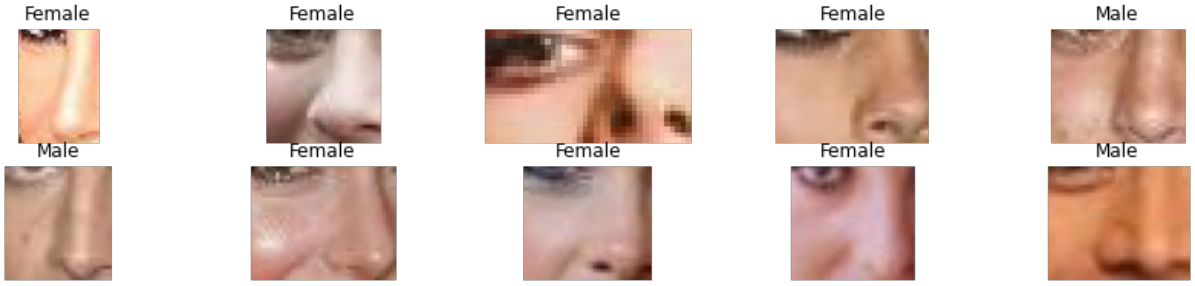

Figure 19

From *Figure 19*, the cropped images of the noses were obtained using Face Landmark Detection with Dlib. However, not all images had their noses detected. Out of the 15000 training images, only 14419 of them detected a nose. And out of the 5000 test images, only 4811 of them detected a nose.

We ran a Stochastic Gradient Descent model on 14419 RGB training images of the noses. The model ran over 1000 epochs, and a mini-batch size of 5. The learning rate is 5 x 10^(-9). The loss history at the end of each epoch and ROC Curve are shown in the plots below in *Figure 20*:

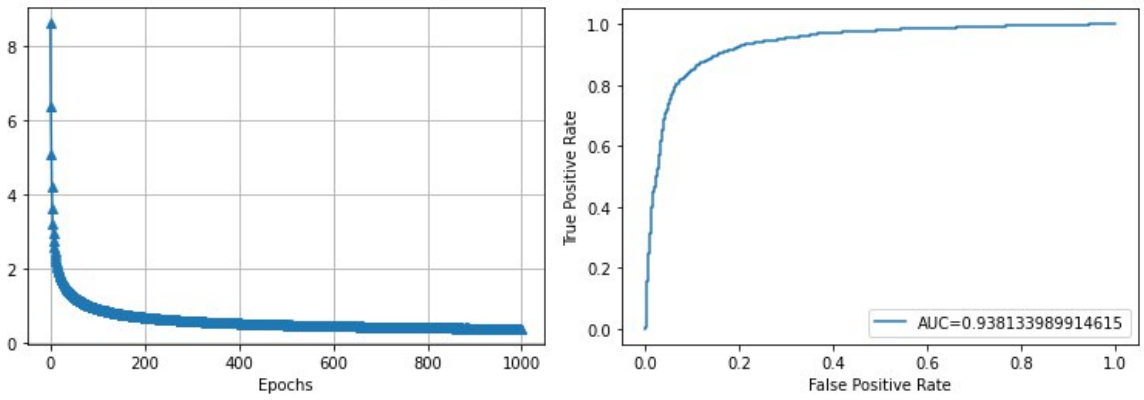

Figure 20

	RGB Images
Training Accuracy	86.3%
Testing Accuracy	86.0%

The model trained on the mouth only is doing almost as well as the model training on the full face. However, its testing accuracy is still slightly lower as compared to the test accuracy of the model on the full face, which has a testing accuracy of 88.7%.

4.4.3 Mouth

Figure 21

From *Figure 21*, similarly, the cropped images of the mouths were obtained using Face Landmark Detection with Dlib. However, not all images had their mouths detected. Out of the 15000 training images, only 14419 of them detected a mouth. And out of the 5000 test images, only 4811 of them detected a mouth.

We also ran Stochastic Gradient Descent model on 14419 RGB training images of the noses. The model ran over 1000 epochs, and mini-batch size of 5. The learning rate is 5 x 10^(-9). The loss history at the end of each epoch and ROC Curve are shown in the plots below in *Figure 22*:

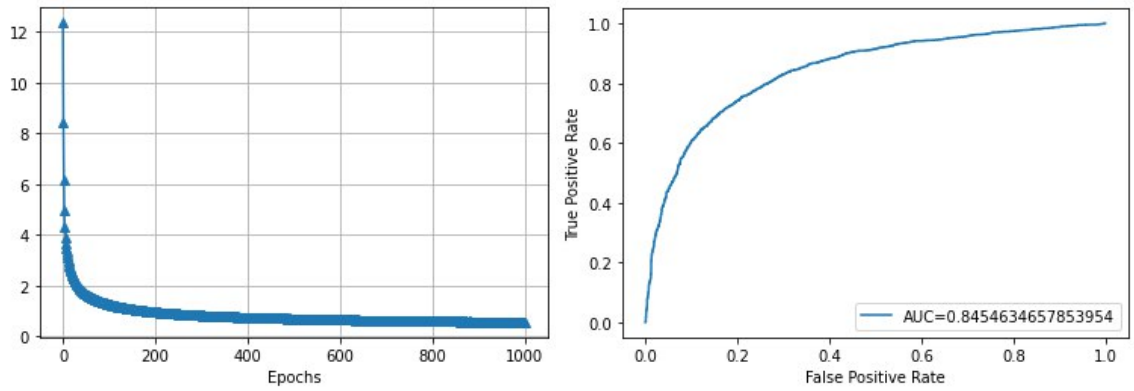

Figure 22

	RGB Images
Training Accuracy	79.2%
Testing Accuracy	77.9%

Although the model trained on noses only is doing reasonably well, its training and testing accuracy, and AUC values are still much lower as compared to that of model trained on the full face and the model trained on the nose.

4.5. Ensemble of these Models (Task 5)

We ensemble the test prediction probabilities obtained from the model trained on noses, mouth and eyes respectively, on their mean probabilities. The testing accuracy obtained was 81.2%, with AUC value of 0.868. The ROC Curve is seen in *Figure 23*:

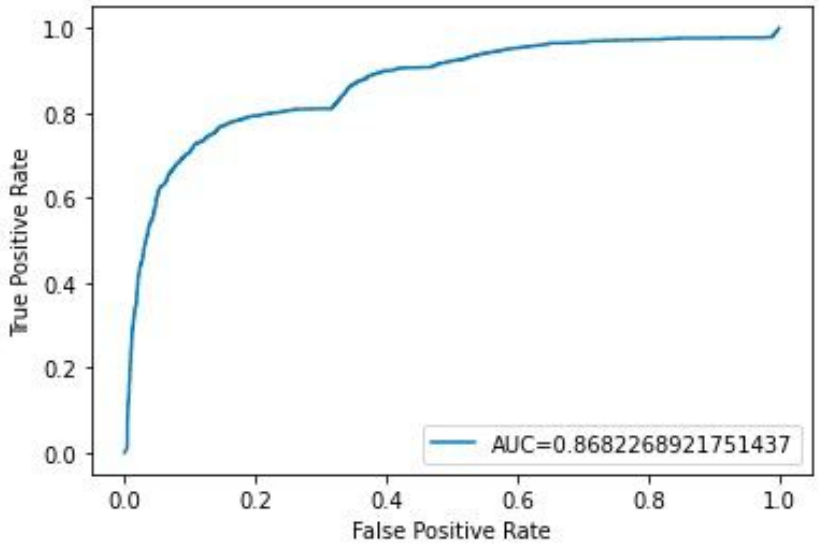

Figure 23

The ensembled model is doing better than the model trained on mouth only, but is doing poorly compared to the model trained on nose and eyes respectively. It is also doing poorly compared to the model trained on the entire image.

It is also important to note that not every image were able to detect at least a mouth, nose and eyes. Hence, we were only able to make predictions on test images which had at least a mouth nose or eyes detected. Thus, the ensemble method may not be the most useful and best model.

4.6 Best Models (Task 6)

From the results above, we saw that using RGB images with an increased contrast intensity would perform better.

However, we hypothesized that removing the background can reduce the image noise, allowing the algorithm to learn the difference in facial features of males and females.

In this section, we would compare between two models: training the model with increased intensity, and training the model with only the front face of an individual.

4.6.1 Increasing contrast

We also experimented with RGB images on their original size, and with increased contrast intensity of 1.9, which performed better as illustrated previously in *Figure 13*. Training the 15000 images on Stochastic Gradient Descent with 1000 epoch, mini-batch size of 5 and learning rate of 5×10^{-9} . The model's accuracy peaked at epoch 400 with testing accuracy 88%, which is still lower than that of Stochastic Gradient Descent run on 15000 RGB images resized to 100×100.

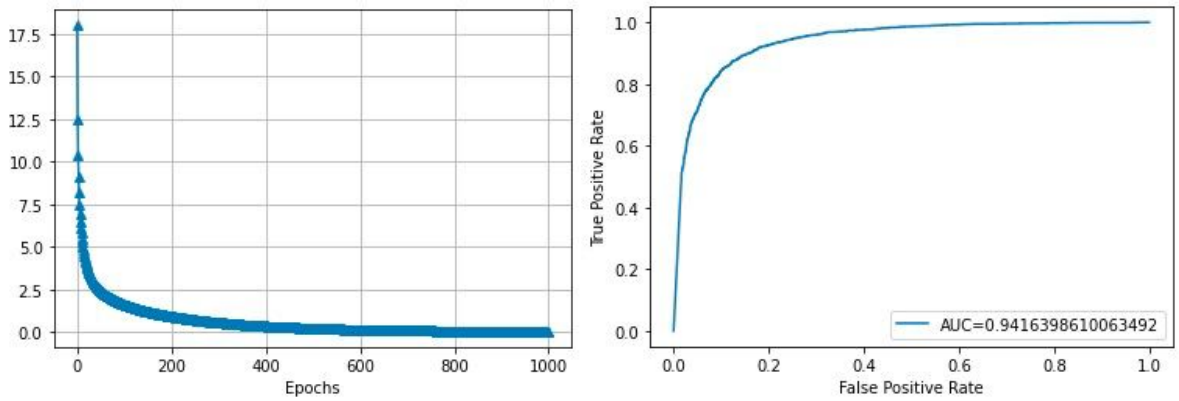

Figure 24

4.6.2 Using Front Face

The front faces of the images were detected using Haar Feature-based Cascade Classifier for Object Detection which is included in OpenCV. We used the pre-trained classifier "haarcascade_frontalface_default.xml" to detect the front face. Out of the 15000 training images, 14121 of them contain front faces. Out of the 5000 test images, 4734 of them contain front faces. Previously, we showed that having around 5000 RGB images or 7000 grayscale images was sufficient to reach an accuracy that is close to that of 15000 images. Hence we believe that this is still a large enough dataset.

An example of a front face detected is as below.

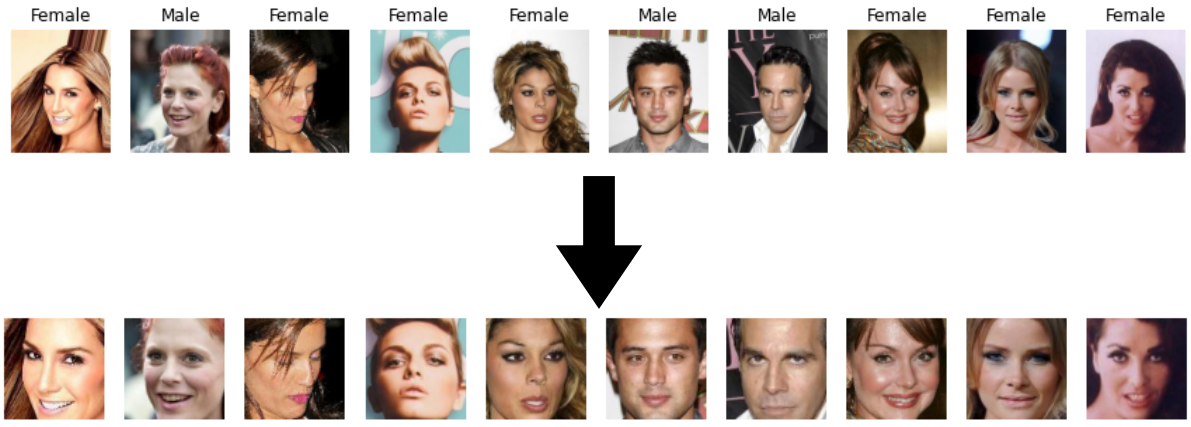

Figure 25

We will first consider grayscale images. A logistic regression model using Stochastic Gradient Descent was used on the front faces dataset.

4.6.2.1 Stochastic Gradient Descent (Grayscale Images)

In this model, we ran the model over 1500 epochs and a mini-batch size of 5. The initial learning rate is set to be 5×10^{-6}. The loss history at the end of each epoch and ROC Curve are shown in the plots below in *Figure 26:*

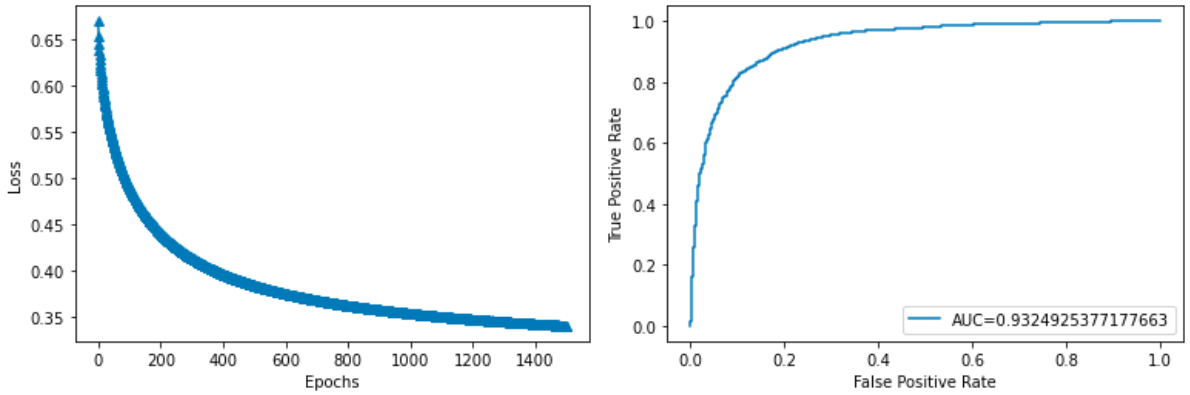

Figure 26

4.6.2.2 Stochastic Gradient Descent (RGB Images)

In this model, we ran the model over 1500 epochs and a mini-batch size of 5. The initial learning rate is set to be 5×10^{-6}. The loss history at the end of each epoch and ROC Curve are shown in the plots below in *Figure 27*:

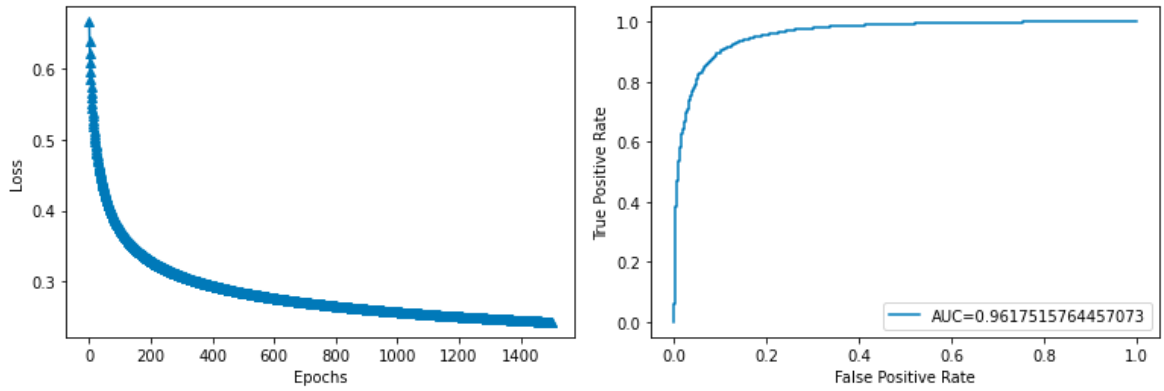

Figure 27

	Grayscale Images	**RGB Images**
Training Accuracy	85.9%	90.5%
Testing Accuracy	86.6%	90.0%

The best test accuracy for grayscale images and RGB images are 86.6% and 90% respectively, These are higher than if we were to use the original images. This could be due to the fact that by limiting the images to only contain faces, we eliminate other noise features such as items in the background.

In conclusion, among all the models, the model that obtained the best accuracy was the model that trained on all 14121 RGB front face images using Stochastic Gradient Descent optimization algorithm with 1500 epochs, mini-batch size of 5 and learning rate of 5×10^{-6}. A test accuracy of 90% was achieved by the model.

4.7 Using only 1% of the data (Task 7)

4.7.1 Using 200 Images

Based on our previous insights, we derived that using RGB images and keeping the highest resolution provided the best test accuracy. Moreover, as we now only use 1% of the data (i.e. the first 200 images), the time taken to train the model would be less of a concern. Hence, we chose to keep the first 200 images as RGB with the original 218 by 178 resolution.

We used Stochastic Gradient Descent as our optimization algorithm on our basic logistic regression model. Using only the first 200 images to train the model, we faced the problem of insufficient data and overfitting. Hence, as shown in *Table 4* below, the model performed poorly when tested on the last 5000 images.

	Basic Logistic Regression
Training Accuracy	100.0%
Testing Accuracy	67.9%

Table 4

In the next section, we explore the effects of data-augmentation and regularization strategies.

4.7.2 Data-Augmentation Strategies

We performed data augmentation strategies to generate replicates of the 200 images by distorting each image in a natural way that human recognition is unaffected. Some data augmentation strategies implemented were horizontal flips, rotations between [-20, 20] degrees and blurring. Using data augmentation strategies, we were able to increase the training set considerably.

Data Augmentation Strategies / Number of training images	None (200)	Flip (400)	Flip, Blur (600)	Flip, Blur, Rotate Clockwise (800)	Flip, Blur, Rotate Clockwise and Anticlockwise (1000)
Training Accuracy	100.0%	100.0%	100.0%	100.0%	100.0%
Testing Accuracy	67.9%	73.2%	73.8%	73.8%	74.0%

Table 5

As shown in *Table 5*, we performed data augmentation strategies on the first 200 images to increase the training set to 1000, resulting in a significant improvement in our testing accuracy. However, we still face the problem of overfitting.

4.7.3 Regularization Strategies

We investigated the effect of regularization strategies on the model, namely ridge logistic regression and lasso logistic regression.

Ridge logistic regression: L^2 penalty term is added

$$\beta_* = \mathbf{argmin}\left\{\beta \mapsto \sum_{i=1}^{N} \mathbf{Loss}(\beta, x_i, y_i) + \lambda\|\beta\|_2^2\right\} \text{ where } \|\beta\|_2^2 = \beta_1^2 + \beta_2^2 + \ldots + \beta_p^2$$

Lasso logistic regression: L^1 penalty term is added

$$\beta_* = \mathbf{argmin}\left\{\beta \mapsto \sum_{i=1}^{N} \mathbf{Loss}(\beta, x_i, y_i) + \lambda\|\beta\|_1\right\} \text{where } \|\beta\|_1 = |\beta_1| + |\beta_2| + \ldots + |\beta_p|$$

Using a range of lambda values, we trained the model on the first 200 images and tested the model's accuracy on the last 5000 images.

Using a lambda value of 5 and 100 on ridge and lasso logistic regression respectively, we obtain the best training and testing accuracies shown in *Table 6*.

	Ridge Logistic Regression	Lasso Logistic Regression
Training Accuracy (best)	100.0%	100.0%
Testing Accuracy (best)	75.5%	76.0%

Table 6

Both ridge and lasso logistic regression have comparable training and testing accuracies. Since the ridge logistic regression model has a lower loss than the lasso logistic regression model, we chose to use the ridge logistic regression model with $\lambda = 5$ as our regularization strategy. Although there may still be overfitting despite the use of regularization strategies, the model performed better on unseen data when a penalty term is imposed on the beta values.

4.7.4 Best Model

Thus, we implemented regularization, namely ridge regression, with $\lambda = 5$ on the 1000 images derived from data augmentation. Besides, we explored using different mini-batch size and number of epochs. We found that having a mini-batch size of 100 gives the best results. Additionally, we concluded that training the model on 800 epochs is sufficient and increasing the number of epochs beyond 800 may lead to overfitting. Our best model achieved a test accuracy of 81.0% on the last 5000 images.

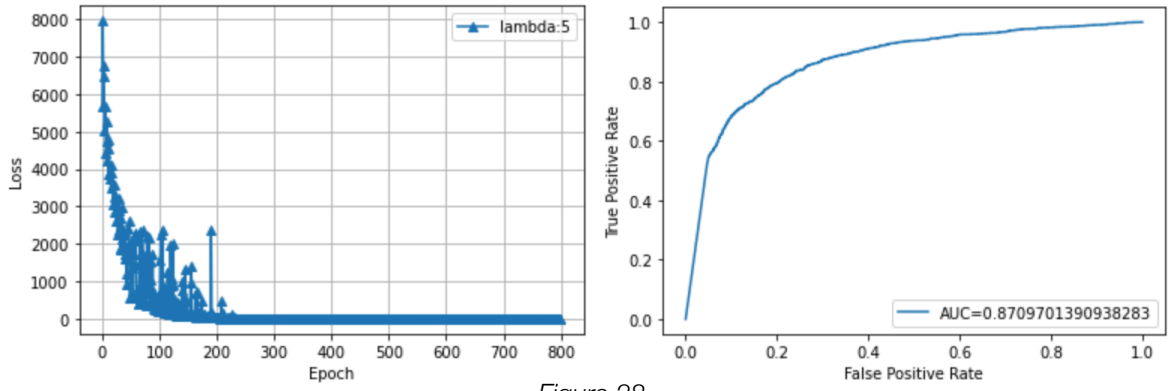

Figure 28

Training Accuracy	100%
Testing Accuracy	81.0%
Area Under Curve	0.871

5 Conclusion

To conclude, when evaluated on the last 5000 images, our best model using all 15000 images achieved a test accuracy of 90.0% and an AUC of 0.962. In comparison, our best model using only 1% of the data (i.e. the first 200 images) achieved a test accuracy of 81.0% and an AUC of 0.871.