

**AY 2021/2022, Semester 2**

**DSA4212 Optimisation for Large-Scale Data-Driven Inference**

Group 9

Lee Wei Qing (A0205666B)

Madeline Lim Chia Bing (A0205053W)

Mabel Lee Wei Ling (A0204397B)

Wu Weiye (A0200578H)

**Table of Content**

[**1 Abstract**](#_w6yirbeefc3v) **2**

[**2 Logistic Regression Models**](#_zcawg3snwkc) **2**

[2.1 Hyperparameters of Baseline Models](#_rf4e3gw5vio5) 2

[2.2 Baseline Models’ Results](#_hzcr041uissu) 2

[**3 Training Sample Size**](#_21oxe5fd308w) **3**

[**4 Resolution of input images**](#_j9m3f579p37s) **4**

[**5 Preprocessing of input images**](#_eq9vysrah4vp) **5**

[**6 Isolation of Features**](#_qvqx004pzfwr) **7**

[6.1 Isolation of Individual Features](#_lg1apwy94svx) 7

[6.2 Ensemble model method](#_1tllxq1e41bx) 8

[**7 Reduced train dataset**](#_tnr66aq5r8t5) **8**

[7.1 Baseline logistic regression model](#_gf15rt2svtch) 8

[7.2 Regularisation](#_n2pa2ezcr6ov) 9

[7.3 Training with preprocessed images](#_rqnthbp2zop5) 10

[7.4 Ensemble methods](#_dd5czvb1gcdb) 10

[**8 Conclusion**](#_sswkzbp3lrff) **11**

[**9 References**](#_cjqjw6lwtjn9) **11**

## 1 Abstract

In this report, we discuss the different models and varying formats of image inputs we explored, and our findings. By running different models on our base pictures, which we standardised in size, we found that the Stochastic Gradient Descent with momentum (SGDM) gives us the best accuracy comparatively. Utilising our best model, we conducted experiments by training on images that were pre-processed (e.g blurring, sharpening), had differing resolutions and cropped/isolated features of the image (e.g eyes, mouth) to study their effects on the model’s accuracy. We also investigated the effects of employing ensemble models with varying combinations of inputs and the use of smaller training dataset (consisting 200 images) on model accuracy.

## 2 Logistic Regression Models

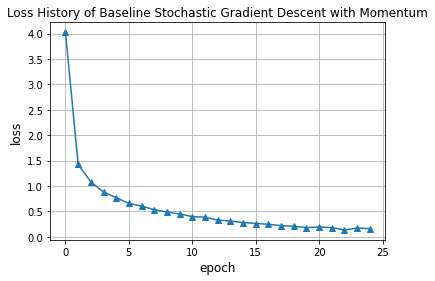
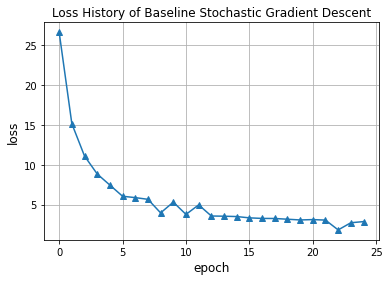
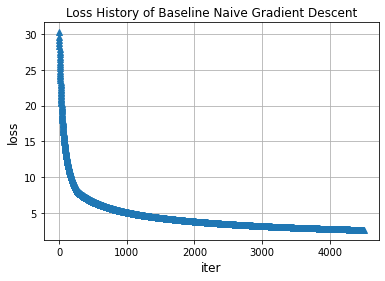
We built the baseline logistic regression models employing 3 different algorithms - Naive Gradient Descent (GD), Stochastic Gradient Descent (SGD) and Stochastic Gradient Descent with momentum (SGDM). These models are trained on the train dataset (first 15000 images) and tested on the test dataset (last 5000 images). The images are resized to a default resolution of 100 x 100.

### 2.1 Hyperparameters of Baseline Models

For GD, we used backtracking as a strategy for automatic step-size selection, with α = 0.5 and trained for 3000 iterations. For SGD, we fixed the learning rate at 0.000001 and trained with a minibatch size of 50 for 25 epochs. For SGDM, we used the same parameters as SGD but introduced β = 0.9 as momentum.

### 2.2 Baseline Models’ Results

The three models took around the same amount of time to train. SGD methods were more computationally efficient and the loss converges much more quickly as compared to GD (refer to Fig. 1). Both SGD and SGDM outperformed GD by a large margin with test accuracies close to 0.90 (refer to Table 1). However, SGDM took significantly fewer epochs to converge and was more stable without major spikes in loss while training. With SGDM, we could achieve similar results within 10 epochs which reduces the training time by more than 50%. Therefore, we chose SGDM as the underlying algorithm for building our subsequent models.



*Fig 1: Plot of the loss history for Gradient Descent (left), SGD (middle), SGD with Momentum (right)*

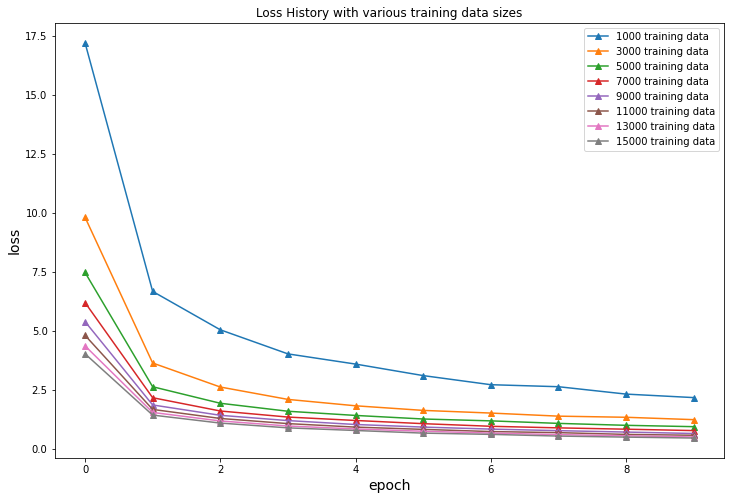
| Method | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| GD | 0.822 | 0.812 | 0.807 |
| SGD | 0.905 | 0.894 | 0.885 |
| SGDM | 0.902 | 0.877 | 0.873 |

*Table 1: Training accuracy, Test accuracy and Test AUC for the 3 baseline models*

## 3 Training Sample Size

We investigated the relationship between the training sample sizes and the accuracy of the model. Training data of various sizes were sampled randomly from the first 15000 images, and evaluated on the last 5000 images. We utilised SGDM with the same hyperparameters as mentioned above in section 2.1. However, the model was only trained for 10 epochs instead of 25. This would greatly reduce the training time without sacrificing too much accuracy since SGDM converges very quickly and 10 epochs were more than enough to obtain a satisfactory result.

Generally, an increasing training sample size would cause the loss to converge much faster (refer to Fig. 2). The increase in rate of convergence also slows down with the increase in training sample size. The test accuracy stabilised when more than 9000 training data were used while time taken to train the data continues to increase with more training data (refer to Table 2).



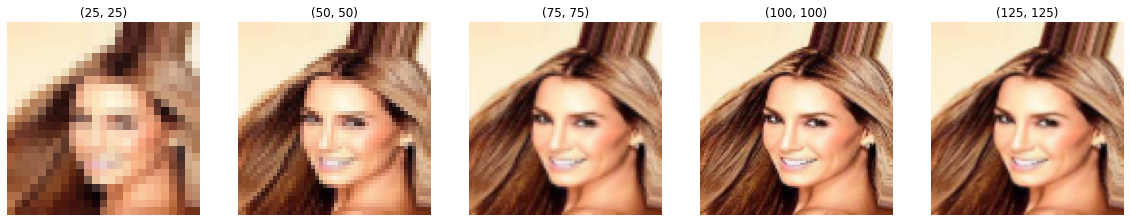
*Fig 2: Plot of the loss history using various sizes of training data*

| Training Size | Training Time (s) | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| 1000 | 73 | 0.827 | 0.807 |
| 3000 | 206 | 0.853 | 0.862 |
| 5000 | 345 | 0.886 | 0.885 |
| 7000 | 482 | 0.893 | 0.891 |
| 9000 | 619 | 0.860 | 0.838 |
| 11000 | 763 | 0.850 | 0.865 |
| 13000 | 910 | 0.901 | 0.895 |
| 15000 | 1035 | 0.891 | 0.897 |

*Table 2: Training time (s), Test accuracy and Test AUC using training samples of different sizes*

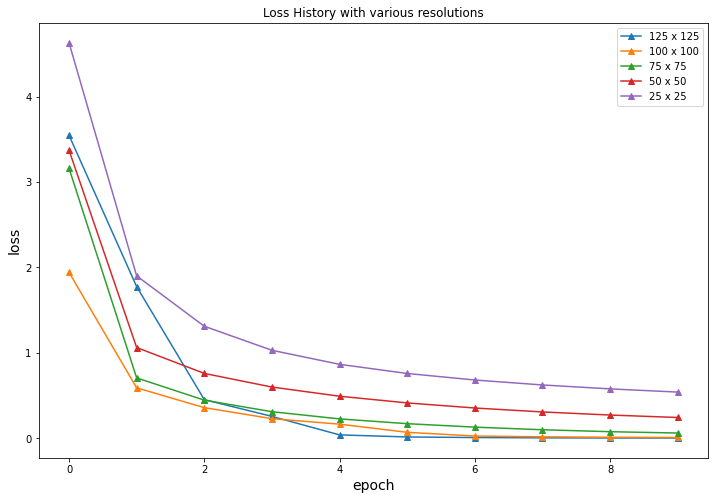
## 4 Resolution of input images

We looked into how the change in resolution of the images will affect the test accuracies. The images were resized to 5 different resolutions (refer to Fig. 3).



*Fig 3: Image 000001.jpg at different resolutions*

After resizing, we proceed to train the model with the images at various resolutions. We sampled 11000 images from the training set for this task to save computational time and memory. With reference to Table 3, higher resolution images will produce better accuracy, however, the improvement is diminishing. Using images of higher resolution will increase the training time exponentially. Therefore, we decided to use the default resolution of 100 x 100 for the subsequent models because the model is able to predict accurately while maintaining a reasonable training time.



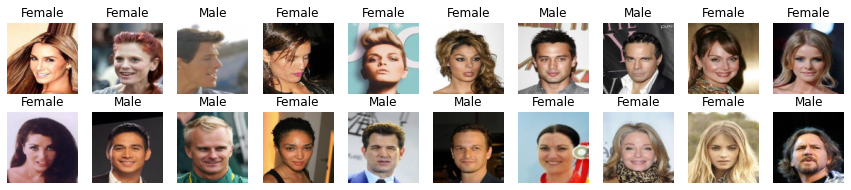
*Fig. 4: Plot of Loss History with various resolutions*

| Resolution | Test accuracy | Test AUC |
| --- | --- | --- |
| 25x25 | 0.863 | 0.857 |
| 50x50 | 0.890 | 0.887 |
| 75x75 | 0.888 | 0.884 |
| 100x100 | 0.889 | 0.885 |
| 125x125 | 0.894 | 0.891 |

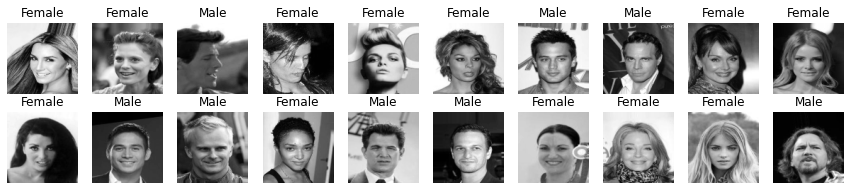
Table 3: Plot of Loss History with various resolutions

## 5 Preprocessing of input images

We experimented training the logistic regression model on pre-processed images to determine if changes made would improve the prediction performance on the test dataset. With the application of Opencv and Pillow libraries, we implemented the image preprocessing methods which include changing image colour (from RGB images) to black and white scale images, blurring, sharpening and changing contrast of image (refer to Fig. 5 a - d).



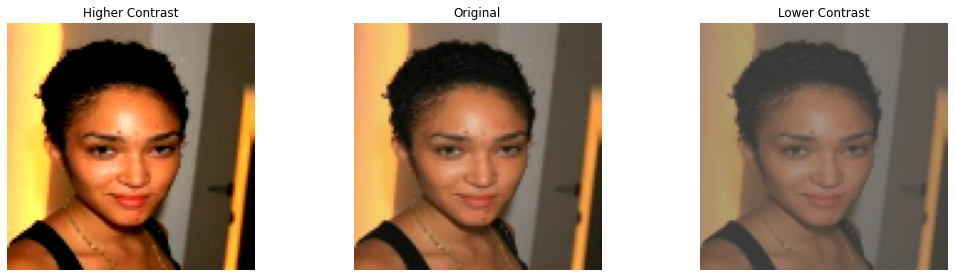
*Fig. 5a: Showing the first 20 image in its original (RGB) form and its labels (male or female)*



*Fig. 5b: Showing the first 20 images in black and white scale and its labels (male or female)*



*Fig. 5c: Showing difference between original (left), sharpened (middle) and blurred (right) image*



*Fig. 5d: Showing difference between Higher contrast (left), original (middle) and lower contrast (right) image*

Subsequently, we trained these augmented images using the logistic regression model with SGD with momentum and found that training with coloured images was indeed necessary as compared to black and white scale images as it results in a higher AUC and prediction accuracy. However, the improvement in the model when trained with images that employ other preprocessing methods (e.g. blurring, sharpening, and change in contrast) is marginal as compared to training on the original coloured image itself. Therefore, we conclude that training the model with the original image has provided sufficient information for the model to predict very well and thus augmented images would not contribute to any further significant increase in prediction accuracy.

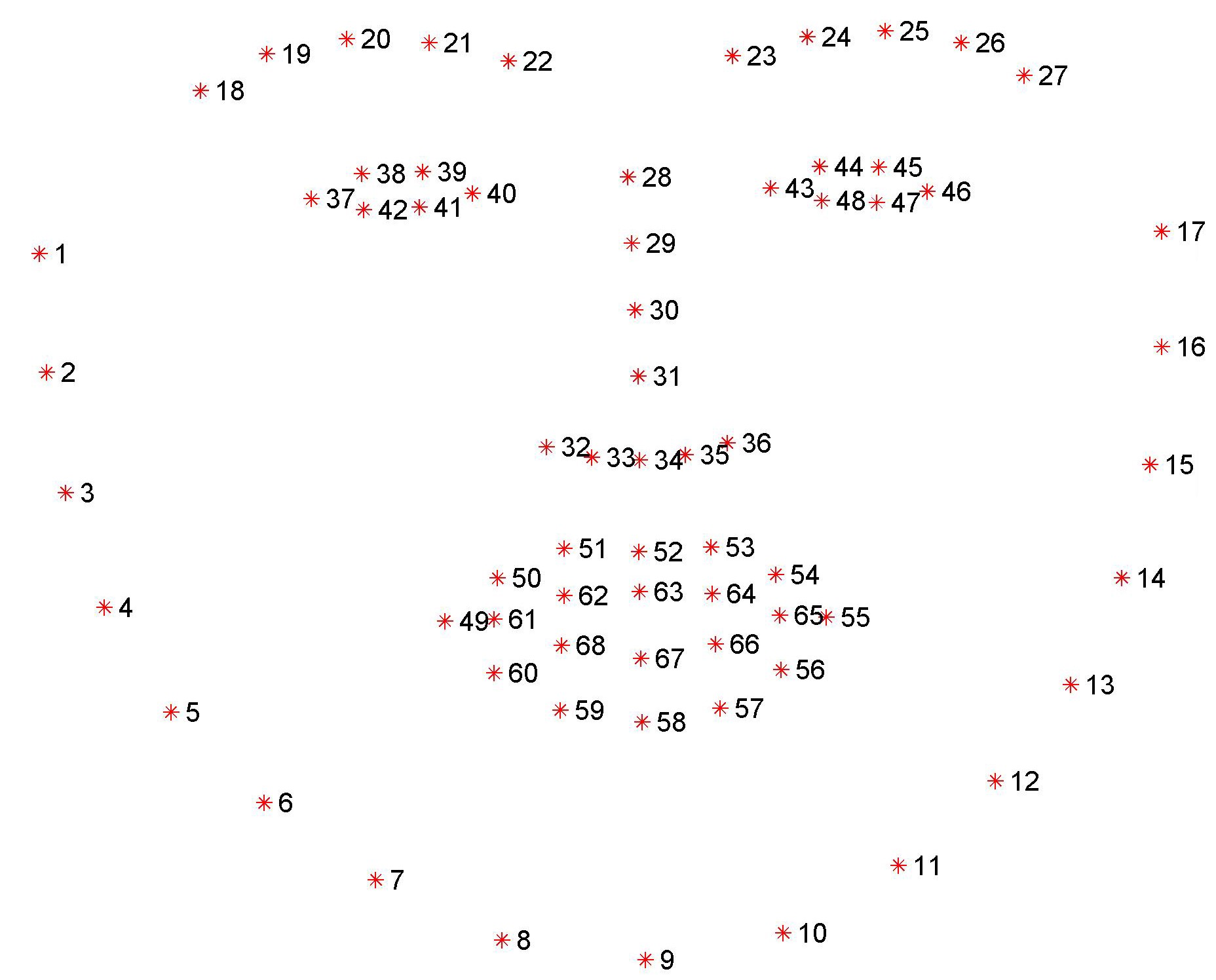
| Images used for training | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| Original (RGB images) | 0.914 | 0.889 | 0.875 |
| Black and white | 0.887 | 0.861 | 0.856 |
| Blurred | 0.900 | 0.889 | 0.875 |
| Sharpened | 0.849 | 0.889 | 0.875 |
| Higher Contrast | 0.962 | 0.883 | 0.886 |
| Lower Contrast | 0.874 | 0.883 | 0.886 |

*Table 4: Results of training with pre-processed images*

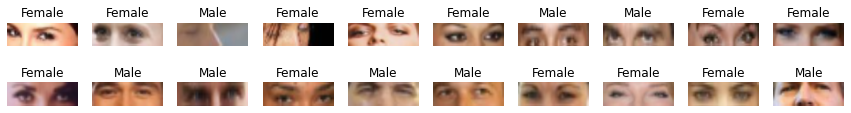
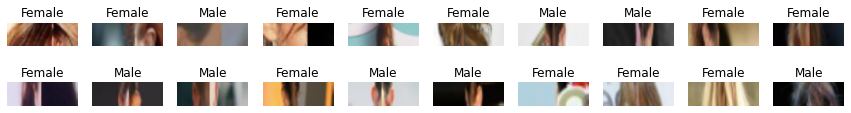
## 6 Isolation of Features

### 6.1 Isolation of Individual Features

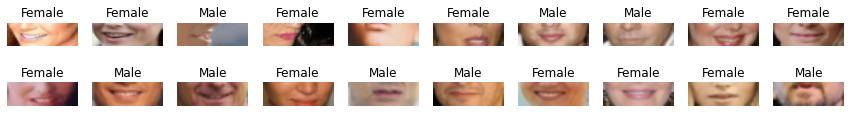
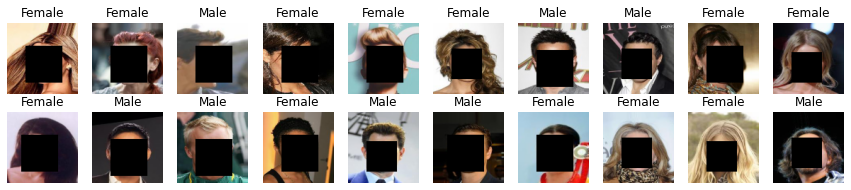
We employed a pre-trained landmark detection model [1], which utilises OpenCV and Dlib. It gives us the coordinates of the 68 landmarks on the face which outlines the facial features (refer to Fig. 6). Using that we selected appropriate landmarks to map out the location of each feature, and naively cropped them. The features we decided to focus on are the eyes, ears, mouth and hair (refer to Fig. 7a - 7d).



*Fig. 6: Image of the 68 landmarks detected by pre-trained model [1]*

*Fig. 7a: Cropped Eyes Fig. 7b: Cropped Ears*

*Fig. 7c: Cropped Mouth Fig. 7d: Cropped Hair*

As mentioned above, for the eyes, mouth and ears, we naively cropped the boundary box for the features by pre-selecting appropriate landmarks from the 68 points, finding out the corresponding (x,y) coordinates and selected the minimum and maximum (x,y) coordinates to map out the boundary box. Specifically for the ears, we concatenated the left and right ears as an image together.  
  
As for the hair, since our pre-trained detector returns us a boundary box which detects the face’s location, and our images are all resized to (100,100), we simply naively selected the facial coordinates to be set as 0 such that it blackens out all facial features, and the hair is preserved in the image.

After storing the isolated features as new arrays, we trained SGDM on images that are cropped to only consist of individual parts of the faces, like eyes, mouth, ears and hair. On an individual level, training models with eyes and hair have performed moderately well with test accuracies of 0.769 and 0.716 and Test AUC of 0.786 and 0.670 respectively (refer to Table 5). However, training models with images cropped at regions with mouth and ears have poor performance with accuracy that are close to the probability of coin flip.The table below represents our exact findings of how the model accuracy varies with facial features.

| Feature Isolated | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| Eyes | 0.781 | 0.769 | 0.786 |
| Mouth | 0.421 | 0.422 | 0.489 |
| Ears | 0.435 | 0.441 | 0.490 |
| Hair | 0.73 | 0.716 | 0.670 |

*Table 5: Results of training with isolation features individually*

### 6.2 Ensemble model method

We then ensemble the model by first making predictions using models built based on each isolated feature. Following which, we aggregated the predictions and took the majority class as our predicted output of an ensemble model. It was found that the best ensemble model which gives the highest accuracy performance on the test dataset is achieved from the combination of the results of the models that were trained on images cropped at region with eyes, mouth and hair, with test accuracy of 0.76 and test AUC of 0.78 (refer to Table 6). However, this ensemble model does not perform as well as our best performing model which trains on the full image itself and has test accuracy of 0.89 and test AUC 0.88.

| Ensemble model | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| Eyes, mouth, hair | 0.779 | 0.762 | 0.777 |
| Eyes, ears, hair | 0.773 | 0.757 | 0.763 |
| Eyes, mouth, ears, hair | 0.765 | 0.751 | 0.751 |

*Table 6: Selected few ensemble model that performs moderately well*

## 7 Reduced train dataset

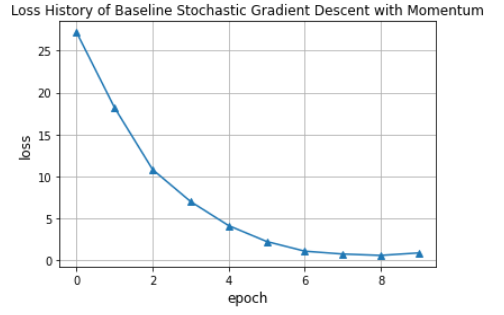
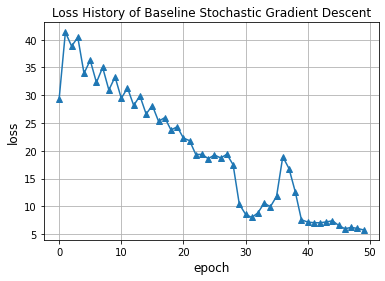
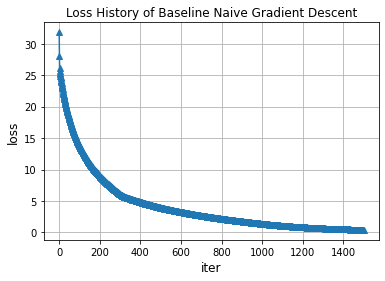
### 7.1 Baseline logistic regression model

As the final task, we used only the first 200 images to train our logistic regression models with the three methods in section 2, namely, GD, SGD and SDGM. For GD, we used backtracking as a strategy for automatic step-size selection, with α = 0.5 and trained for 1500 iterations. For SGD, we fixed the learning rate at 0.000001 and trained with a minibatch size of 50 for 50 epochs. For SGDM, we used the same learning rate as SGD at 0.000001 but introduced β = 0.9 as momentum. With a minibatch size of 50, the model was able to converge quickly using 10 epochs.

We observed that the logistic regression model with SGDM still performs the best with test accuracy of 0.688 and test AUC of 0.699 albeit a lower prediction performance as compared to training with 15000 images (refer to Table 7). Similar to results obtained in Section 2.1, SGDM took fewer epochs to converge as compared to SGD. Unsurprisingly, there were signs of overfitting as the train accuracy tends to be considerably larger than the test accuracy, suggesting that the model is closely fitted to the training dataset. This phenomenon is due to the usage of a relatively small dataset of training images.

| Method | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| GD | 0.795 | 0.667 | 0.657 |
| SGD | 0.820 | 0.656 | 0.677 |
| SGDM | 0.905 | 0.688 | 0.699 |

*Table 7: Training accuracy, Test accuracy and Test AUC for the 3 baseline models*

**

*Fig 8: Plot of the loss history for Gradient Descent (left), SGD (middle), SGD with Momentum (right)*

### 7.2 Regularisation

As we are training with a smaller data set, we see that overfitting tends to occur. Hence, we experimented with regularised versions of logistic regression to mitigate overfitting.

For ridge regression, we used λ = 0.5 and for lasso regression, we used λ = 0.4. We observed that fitting regularised versions of logistic regression models with SGDM do not help to prevent overfitting entirely however, we see that the test accuracy of the model fit with lasso regularisation improved the test accuracy from 0.688 to 0.719 and test AUC from 0.699 to 0.710.

| Regularisation Method | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| Ridge | 0.900 | 0.721 | 0.690 |
| Lasso | 0.990 | 0.719 | 0.710 |

*Table 8: Training accuracy, Test accuracy and Test AUC for the regularised versions of models*

### 7.3 Training with preprocessed images

We also experimented training with preprocessed images using the lasso-regularised logistic regression model with SGDM. We observed that training on blurred images improved the test accuracy and test AUC for reduced train dataset. Blurring is a technique which helps to reduce the amount of noise in the image. Hence, it works in ensuring that the model fitted on 200 images does not fit too closely to the noise and instead, is able to generalise well to other images.

Preprocessing methods such as lowering the contrast, sharpening and use of black and white images resulted in much poorer performance while higher contrast images seem to produce similar results as the original coloured images (refer to Table 9).

| Images used for training | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- |
| Original (RGB images) | 0.990 | 0.719 | 0.710 |
| Black and white | 1.000 | 0.667 | 0.647 |
| Blurred | 1.000 | 0.751 | 0.735 |
| Sharpened | 0.825 | 0.672 | 0.660 |
| Higher Contrast | 1.000 | 0.730 | 0.713 |
| Lower Contrast | 0.825 | 0.661 | 0.682 |

*Table 9: Results of training the 200 preprocessed images*

### 7.4 Ensemble methods

We then build ensemble models by first making predictions using models built based on each of the preprocessed images. Following which, we aggregated the predictions and took the majority class as our predicted output of an ensemble model.

Since we have already identified SGDM with lasso to be a model that outperformed other methods, it is the only model we took into consideration. We trained SGDM with lasso on each of the preprocessed images, and made predictions using the training and test set. Then, we then identified all unique combinations of variables among the five preprocessed images we have. We highlighted the various ensemble models that did well in test sets in the table below (refer to Table 10).

| Number of features | Black White | Blur | Sharp | High  Contrast | Low  Contrast | Train accuracy | Test accuracy | Test AUC |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 |  | ╳ |  | ╳ | ╳ | 1.000 | 0.744 | 0.737 |
| ╳ | ╳ |  |  | ╳ | 1.000 | 0.734 | 0.731 |
| ╳ | ╳ |  | ╳ |  | 1.000 | 0.747 | 0.729 |
| 4 | ╳ | ╳ |  | ╳ | ╳ | 1.000 | 0.746 | 0.728 |
| ╳ | ╳ | ╳ |  | ╳ | 1.000 | 0.735 | 0.716 |
|  | ╳ | ╳ | ╳ | ╳ | 1.000 | 0.732 | 0.715 |
| 5 | ╳ | ╳ | ╳ | ╳ | ╳ | 1.000 | 0.734 | 0.725 |

*Table 10: Selected few ensemble model that performs moderately well*

Holding these results against our original SGDM model with lasso regularisation that was trained on the entire image, which has test accuracy and test AUC of 0.719 and 0.710 respectively, we can see that ensemble methods perform better than SGM with lasso trained on the original image alone. In particular, the ensemble SGDM model with lasso regularisation that was trained on black and white, blurred and high contrast images respectively, performed the best with test accuracy of 0.747 and test AUC of 0.729.

## 8 Conclusion

In conclusion, we found that training a logistic regression model with Stochastic Gradient Descent with Momentum (SGDM) on coloured images gives us the best prediction performance with test accuracy of 0.89 and test AUC of 0.88 when trained with 15000 images. In addition, the improvement of training with other forms of preprocessed images was marginal. Furthermore, we found that using ensemble methods on isolated features of the image, in particular the eyes, mouth and hair, gives moderately high test accuracy of 0.76 and test AUC of 0.77. We recognised that training with full images performed better and this could be due to the greater amount of information available for the model to learn its underlying patterns.

We have also explored the possibility of training with a largely reduced training dataset of 200 images and found that introducing regularisation to logistic regression with SGDM has improved its test prediction. In addition, the ensemble method with models trained on black and white, blurred and high contrast images respectively, had achieved a moderate test accuracy of 0.747 and test AUC of 0.729. However, working solely with blurred images still reported the best result of test accuracy of 0.751 and test AUC of 0.735 out of all the methods used to improve the model’s performance. Ultimately, it is unsurprising that training with reduced dataset gives performance that is less significant since training with small dataset causes the model to fit closely with the noise.

## 9 References

[1] A. Rosebrock (April 3, 2017) Facial landmarks with dlib, OpenCV, and Python. Retrieved on March 21, 2022 <https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>