Spring 2021

Bias and Fairness of Evasion Attacks in Image Perturbation

SiChong Qin
Central Washington University, qins@cwu.edu

BIAS AND FAIRNESS OF EVASION ATTACKS IN IMAGE PERTURBATION

A Thesis

Presented to

The Graduate Faculty

Central Washington University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Computational Science

---

by

SiChong Qin

June 2021

CENTRAL WASHINGTON UNIVERSITY

Graduate Studies

We hereby approve the thesis of

SiChong Qin

Candidate for the degree of Master of Science

APPROVED FOR THE GRADUATE FACULTY

_____	_____
	Dr. Razvan Andonie
_____	_____
	Dr. Boris Kovalerchuk
_____	_____
	Dr. Szilard Vajda
_____	_____
	Dean of Graduate Studies

ABSTRACT

BIAS AND FAIRNESS OF EVASION ATTACKS
IN IMAGE PERTURBATION

by

SiChong Qin

June 2021

When talking about protecting privacy of personal images, adversarial attack methods play key roles. These methods are created to protect against the unauthorized usage of personal images. Such methods protect personal privacy by adding some amount of perturbations, otherwise known as ”noise”, to input images to enhance privacy protection. Fawkes in Clean Attack 4.1 method is one adversarial machine learning approach aimed at protecting personal privacy against abuse of personal images by unauthorized AI systems. In leveraging the Fawkes in Evasion Attack method and through running additional experiments against the Fawkes system, we were able to prove that the effectiveness of perturbations added in privacy protection of images depends on how we stratify the input population based on demographic features such as race and gender, showing that we need to be able to quantify and take into account various potential areas of bias when leveraging adversarial attack methods to ensure optimal protection of all input images.

As it currently stands, the Fawkes system has a fixed set of hyper parameters for amount of perturbations added per image, which essentially means that they consider all users be treated identically in terms of amount of perturbations added. However, from testing our hypothesis through running various experiments, we found that the protection performance is statistically significantly different when the input images are

from different groups of people based on demographic features like race and gender when applying the original parameter settings. For example, we found that for light skin toned females, the original Fawkes settings work well in ensuring privacy protection. However, the original Fawkes settings do not perform well with dark skin toned males in ensuring privacy protection of these images.

In order to ensure fairness from the system, we propose guidelines for taking into account these demographic differences in order to get optimized solution sets for hyper parameter tuning, making future users of the model aware of existing biases and how to mitigate and take them into account. Our proposed solution for hyper parameter tuning takes into account demographic features with internal system settings, aimed at improving the protection performance for all skin tones and gender. We categorized inputs based on demographic features (namely, race and gender) and then used the current Fawkes model to process the categorized input images with different parameters. In our experiments, the main metric we use to evaluate and determine the optimal hyper parameters is the output of custom classifier models (e.g., confidence values) built from Microsoft Cognitive Services Face API .1. From a high-level, we first test the effectiveness of the Fawkes model applied in Evasion Attack Scenario. Then we ran experiments with curated datasets to prove the existence of demographic bias in the current Fawkes model with its default parameters. Next, we performed experiments on changing the default parameters of Fawkes to analyze the influence of different parameters on different input images. Based on the previous experiment results, we propose guidelines and solution sets that optimize the internal settings to ensure Fawkes model takes into account potential demographic biases and ensure fair protection for all input images.

Our proposed solution and devised set of guidelines takes into account various demographic features (e.g., race and gender) and internal settings together by using grid-

search like methods, namely pair-to-pair [1]. By applying our proposed set of guidelines, we ensure optimal protection performance by all skin tones and gender, improving bias and enhancing fairness of the Fawkes model.

ACKNOWLEDGEMENTS

This thesis is dedicated to my grandfather, Shoubin Qin, an incredible lifelong teacher and an even more incredible grandfather, who passed away before he could see my thesis and graduation to fruition. Thank you to my grandfather for his unconditional love, support and lifelong inspiration. Starting from proposal to final formulating thesis, I am grateful to have received tremendous amounts of support and guidance from my family, mentors and advisor. I would like to thank my advisor Professor Andonie who has given me support and invaluable advice on creative methodologies and providing solutions for challenging problems. I also would like to thank the committee members: Dr. Szilard Vajda and Dr. Boris Kovalerchuk for reviewing this thesis and giving me advice and support. I would like to thank my aunt Xiaoli Qin and cousin Connie Yang who have provided me with discussion over high level thinking with logic, their patience and knowledge have helped me grow significantly as a researcher in analyzing problems and designing solutions. I would also like to thank my uncle Jie Yang for discussing high-level theoretical concepts. Lastly, I would like to thank my grandparents Shoubin Qin and Xiuzhen Zhou and parents Xiaofeng Qin, Cuihong Si for their unconditional support and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

From existing research and work [2], Fawkes is applied as an adversarial attack method in the Clean Label Attack .2. Fawkes has shown to achieve high protection rate when tested on mainstream online APIs [2], such as AWS, Microsoft Azure Facial Cognitive Services. Fawkes will be applied in Evasion Attack .2, because we take APIs results as metric to evaluate the effectiveness of perturbations generated by Fawkes, where we first wanted to ensure that Fawkes would perform relatively well.[1]

This thesis has two main goals: 1) conduct additional research on Fawkes settings taking into account added demographic properties like race and gender; 2) extend the Fawkes model such that it takes into account different demographic features of input populations (e.g., race and gender), in order to ensure optimal privacy protection for all. The first goal is to analyze the impact of the main threshold - Structural Dissimilarity [3] (DSSIM .2) together with two extra parameters (Max Iterations and Learning Rate) of the Fawkes model on the different categorized inputs by race and gender under the Evasion Attack. The second goal is to use the optimized parameters from the previous stage to build an extension to Fawkes, under Evasion Attack that considers demographic differences of inputs through our defined categorized input system. Initially, our input categories are race and gender, where we hope to extend this in future works. By categorizing the input population, we take into account potential hidden biases created by the different categories where we aim to improve both the performance (measured

[1]Fawkes has achieved relatively decent performance by using Fawkes original modes(Table.1), the protection of perturbations that Fawkes provides achieve good performance 5.1

primarily by our custom classifier's average output probabilities [4] and efficiency (measured primarily by processing time of Fawkes model 5.5).

Our main contribution is to devise a set of robust guidelines to account for Fawkes model's demographic biases (e.g., race and gender) in the Fawkes system, through analyzing the influence of the Fawkes parameters' settings to capture the stratification of different racial and gender categories, ultimately bringing awareness to and enhancing fairness of the Fawkes model.

As pointed out in one article published [5] last year in the area of racial biases and fairness, assessing racial and gender biases in machine learning classification work is extremely important and critical in ensuring that we are responsibly leveraging intelligent technology and solutions as they become omnipresent in society and daily life.

Our initial question was to evaluate the hypothesis: if under adversarial attack .2, would Fawkes model's performance be statistically significantly influenced by racial and gender characteristics from the input datasets. After preliminary experiments 4.1 in testing this hypothesis, we found there did indeed exist unfairness (e.g., between lighter and darker skin toned people, broadly categorized by race), which implies that the Fawkes model should not treat inputs the same in order to maximize privacy protection if under adversarial attack by unauthorized people or companies.

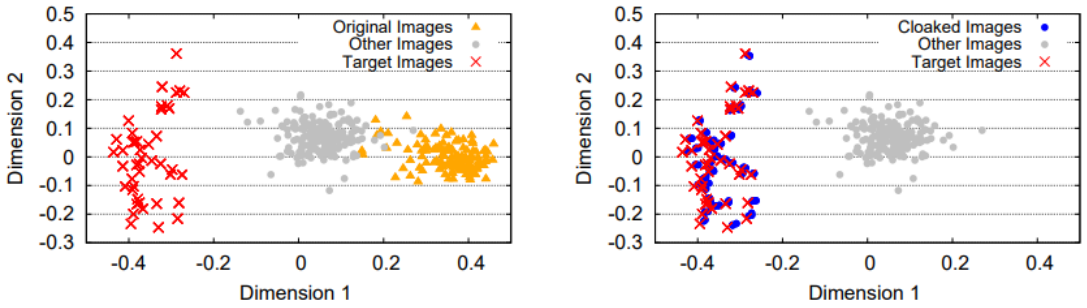
CHAPTER II

RELATED WORK: FAWKES

Fawkes is an adversarial attack method that helps users to modify their personal images against unauthorized usage like facial recognition models [2]. The scenario in Fawkes is that hackers or unauthorized third parties may use users' personal images to build their facial recognition system, by applying methods (like Scrapy) to collect images automatically. When users upload their images to social medias, the images will be processed by Fawkes, adding perturbations to the uploaded images.

Once the hackers use these modified images to train their facial recognition model, the modified images will affect their model's performance. Fawkes, the system we used to process images, will merge one candidate's facial features with the inputs. As stated in [2], the final results can be understood as Feature Space Deviation. The Fawkes team uses PCA [6] to describe their conclusion as shown in Fig.1.

The modification of images of Fawkes applies a specific algorithm to add perturbations to the original input image. The perturbations are based on the features of Target Class and Input Image. Fawkes has four different modes that control levels



(a) Before Cloaking (b) After Cloaking

FIGURE 1: Feature Space Deviation [2]

of perturbations: Min, Low, Mid, and High Mode (Version 0.3.2). DSSIM, measuring structural dissimilarity [3], functions as the main threshold of the perturbation level. By leveraging the DSSIM value, Fawkes can apply the different levels of perturbations to input images. Apparently, DSSIM is the main threshold. There are also some variables engaging to solve the optimization problem that calculates the perturbations.

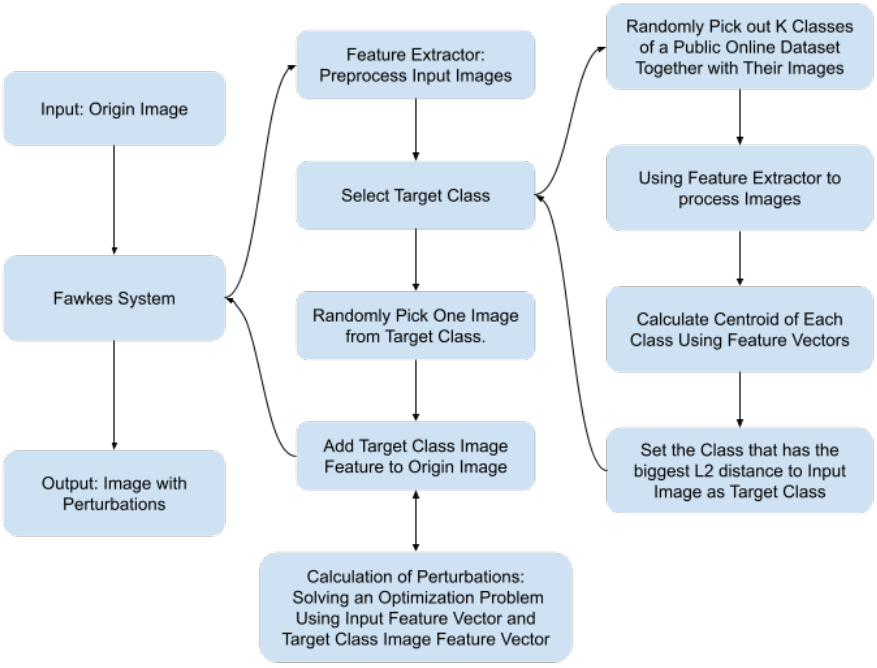


FIGURE 2: Fawkes Working Flow

When an image is input to Fawkes, it will go through the following steps (see Fig.2):

1. Transform image into feature vectors by using feature extractor built by applying transfer learning (a pre-trained model together with a dataset).
2. Find targeted class as candidate. In target selection, it also uses feature extractor to process the candidate image. Since it involves randomness in target selection, the processed images of the same original images have a high probability of being different from each other.

3. In the target class, it randomly pick one images as candidate image for calculating perturbations.
4. Use the images get in step 3 combining with input images to solve an optimization in order to get optimal perturbations.
5. Merge the perturbations with original input image and then output.

The perturbations are calculated by solving an optimization problem (Eq.2.1),

$$\begin{aligned} &\min_{\delta} && Dist(\phi(x_T), \phi(x \oplus \delta(x, x_T))) \\ &\text{subject to} && |\delta(x, x_T)| < \rho \end{aligned} \tag{2.1}$$

where:

- x: Input image (without perturbations).
- x_T: Image from selected candidate class.
- $\delta(x, x_T)$: Perturbations computed for x based on image x_T from label T. The calculation of perturbation is based on current:DSSIM (Structural Dissimilarity).
- $x \oplus \delta(x, x_T)$: Image x with perturbations.
- ϕ: Feature extractor built by applying transfer learning.
- $\phi(x)$: Feature vector x, image x processed by feature extractor.
- $Dist$: Euclidean distance.

There are three parameters that engage in making perturbations in Fawkes: DSSIM, Max Iterations, and Learning Rate. DSSIM value controls the level of perturbations and functions as the main threshold of the system; Max Iterations and Learning Rate hels in solving optimization problem. In order to search for the final results, Fawkes applies

the Gradient Descent algorithm. By applying Gradient Descent, Fawkes will need two variables – Max Iterations and Learning Rate. Max Iterations denotes the maximum times of conducting computation of Gradient Descent. Learning Rate denotes the speed of convergence.

The original Fawkes mode settings are depicted in Table.1. These four modes function as different levels of perturbations. From the table, when increasing the DSSIM value, the perturbations will be strengthened.

The details about how these settings are acquired have not been described in Fawkes [2] paper which are designed and set by Fawkes team.

Mode	DSSIM	Max Iterations	Learning Rate
Min	0.002	20	40
Low	0.003	50	35
Mid	0.005	200	20
High	0.008	500	10

TABLE 1: Original Fawkes Settings.

CHAPTER III

METHODOLOGY

In the following we will describe our proposed methodology and guidelines to determine how much perturbations need to be added to the images leveraging Fawkes model to ensure optimal privacy protection for images of individuals from different demographics, focused primarily on race and gender.

Fawkes can be viewed as a black box with several inputs. The thesis is going to enhance the adversarial attack method Fawkes by building a categorized input system and set of guidelines for future Fawkes model users to take into account potential racial and gender biases while leveraging Fawkes. First, we conducted exploratory data analysis to discover the relationships among input properties and Fawkes settings, where we experiment with the amount of perturbations that needs to be added by varying input images based on race and gender while holding other parameters constant and holding input images constant while varying other input parameters and generally researching the properties of inputs and effects of default Fawkes settings. Ultimately, we will develop a set of recommended optimal parameters through an original methodology for inputs of different race and gender to ensure Fawkes model is optimally protecting all image's privacy against potential unauthorized usage by other machine learning systems.

Intuition about Parameter Optimization

Based on Experiment A 4.2 ran to test our initial hypothesis on whether Fawkes currently performs equally for individuals of all skin tones, we found that for different skin tones, Fawkes' performance is statistically significantly different 4.2. For example, from Experiment A 10, we found that Fawkes performs much better on the Caucasian

than the African female adult, meaning that Fawkes would do a better job at ensuring Caucasian female adult images are privacy protected against adversarial attacks compared with that of African female adult images. In this case, we need to search for more optimal parameters for groups like African Female adult group. The final results we want to achieve is to make the average protection performance of African female adult group as close as to that of Caucasian female adult group. We consider this a parameter optimization problem to find the optimal parameters for each group of people. There are two approaches we can take to achieve this work, where our thesis will focus on leveraging Proof by Exhaustion and the other approach will be described in future works and discussion section.

Evaluation Metric

The key metric we used in evaluating the effectiveness of amount of perturbations need to be added by Fawkes to images in order to protect privacy one of the outputs of classifier, a probability called confidence in the Face Client library from Microsoft Cognitive Service Face APIs [4]. From the Face Client library, we call the identify method that takes an array of detected faces and compares them to a PersonGroup .2. If it can identify a detected face of input image to a Person in PersonGroup, then the model will output a result described by using confidence values. The confidence describes how similar the input to its actual label.

In our scenario, after we add perturbations to the test set images, we put the perturbed images into our trained API model.

We train the API model with unmodified images (the images without processed by Fawkes). We used the perturabated images to acquire the confidence values we will use for the experiments. Higher confidence value denotes that the input image is more

identical to the its actual class/label. Lower confidence value denotes that the input is not identical to its actual class/label.

Computation Procedures

The overall Computation Flow we considered for the experiments and research is described as below.

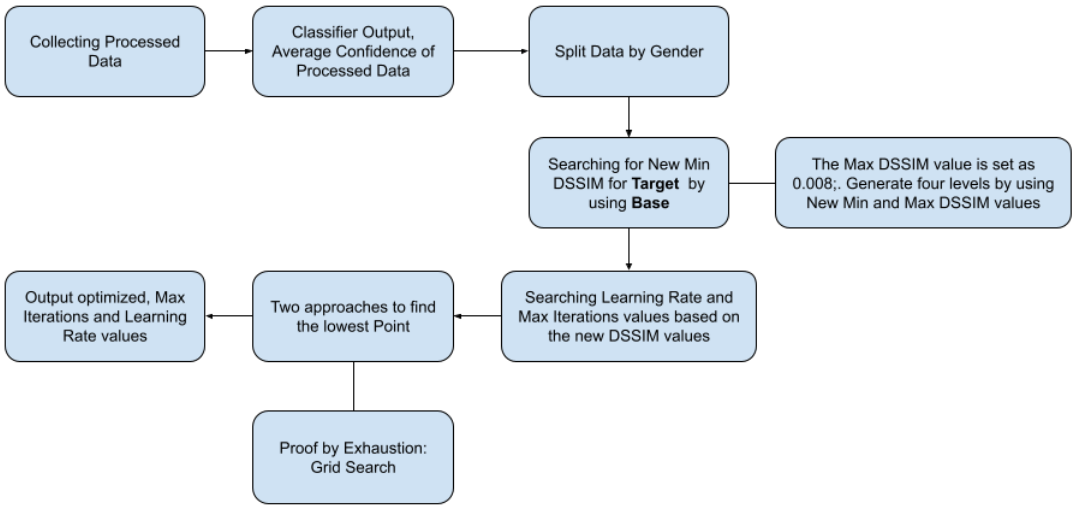


FIGURE 3: New Methodology Description.

First, we use online APIs .1 from Microsoft Cognitive Services to train a facial recognition model with an input dataset consisting of images no perturbations, after dividing the input dataset into train and test sets. This input dataset has 60 distinct individuals (Details in Table.2) where each individual has around 60 different images from different angles, backgrounds and lighting. Second, we use the test set to generate perturbed or modified images by applying the Fawkes model. After we have the APIs trained and perturbed images generated, the trained API will be used to give out the recognition value called confidence [7]. Since we use test set to generate perturbed images, we will use the average confidence of these group of images to represent the

performance of Fawkes. The probability, or confidence, of belonging to a certain class computed by our trained facial recognition model represents how similar any input image is to its actual label class. Please refer to Fig.4 for a visual representation of this procedure.

Record	Gender	Skin Tones	Amount of People
1	Male	Caucasian	10
2	Male	Asian	10
3	Male	African	10
4	Female	Caucasian	10
5	Female	Asian	10
6	Female	African	10

TABLE 2: Pairs for Data Generation.

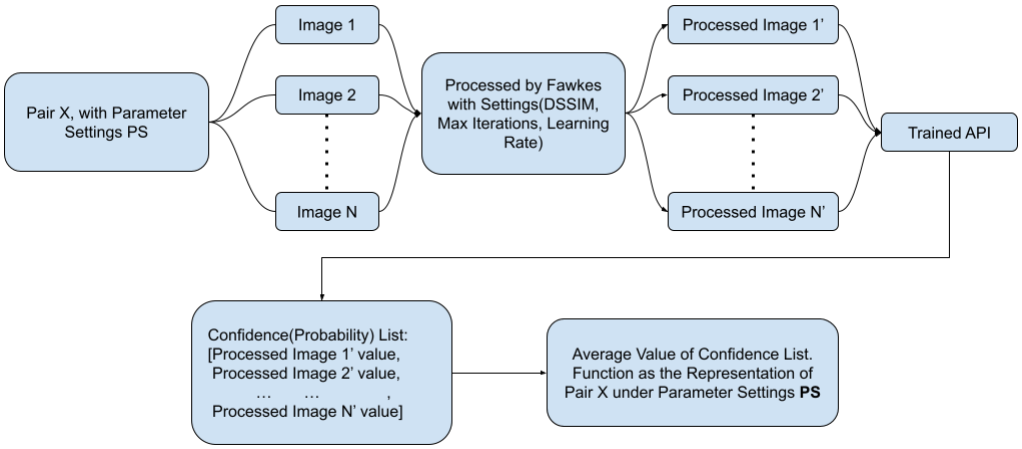


FIGURE 4: How to acquire Confidence and Average Confidence.

For our categorized input, we further divide them into race and gender pairs, as shown in Table.2, denoted by Records 1-6. Each record (or pair) contains images of 5 distinct individuals (randomly extracted from 10 people) where each of those individuals will have 20 distinct images of themselves from different angles, lighting and backgrounds, making a total of 100 images. We chose these initial number of individuals

to optimize for computation and reduce overall computation costs, as processing of each image takes non trivial amount of time 5.5.

CHAPTER IV

EXPERIMENTS

To find the optimal parameters of each different group, we applied the Grid Search algorithm to perform hyperparameter tuning, especially for Max Iterations and Learning Rate. Before leveraging Grid Search [1] for optimal hyperparameter tuning, we first need to prove that Fawkes model is effective in Evasion Attack .2 so that we can use the output of API model, confidence values, as our metric. In Fawkes, it was applied in Clean Label Attack [2], which is different from Evasion Attack. In Clean Label Attack, modified images are mixed with normal unperturbated images in training stage. In Evasion Attack scenarios, modified images are put in testing stage. Considering the differences of these two different scenarios, we think it would be sufficient to prove Fawkes's effectiveness under Evasion Attack.

Preliminary Experiment - Effectiveness in Evasion Attack

In a preliminary experiment, we test Fawkes using Mid mode and High mode 1 under scenario to show that Fawkes can not only work in Clean Label Attack .2 but also Evasion Attack .2. We use the following software platforms and APIs: Microsoft Cognitive Service Face API .1; Multi-task Cascaded Convolutional Networks (MTCNN) [8], a method to detect human faces in images.

Microsoft Cognitive Service Face API is user-friendly, interpretable and proven to be the most effective among its peers in facial recognition services. Microsoft Cognitive Service Face API effectively avoids cropping steps (it combines detection in person group) and avoids massive parameters tuning when training models. In the documentation

of Microsoft Cognitive Service Face API, Section Assign faces to Persons, it is mentioned that it would detect faces and assign to the correct person [4].

MTCNN is used in Fawkes to detect faces in images, since Fawkes need to first find the face then crop the face out of its background and finally add perturbations to the images [2]. Another purpose of applying MTCNN is to filter out the images that do not contain detectable faces.

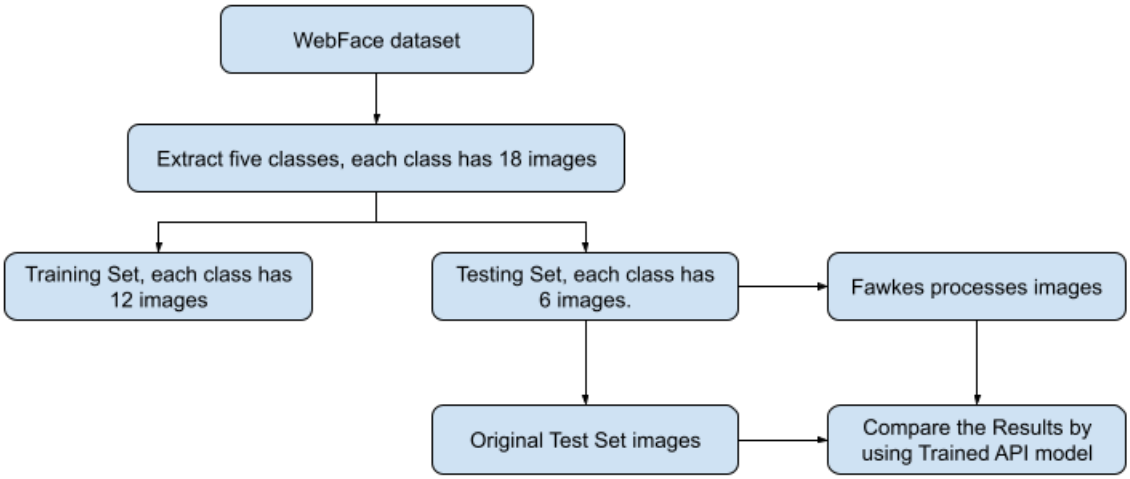


FIGURE 5: Evasion Attack Procedure of Fawkes.

We extract our dataset from the source dataset, CASIA-WebFace dataset [9], a large-scale face recognition dataset with up to 10,000 subjects and 500,000 faces. From there, we curate our dataset to contain 5 different individuals with 18 images per individual where we then take 12 images for training and 6 images for testing into the Fawkes model, mimicking the 60-40 train-test split [2]. This experiment contains three parts where the overall structure is shown in Fig.5. The first part is training a custom facial recognition model with our curated dataset using Microsoft Cognitive Service Face API on our training set. The second part is using the trained model to process testing images using Fawkes under certain modes, specifically Mid-mode and High-mode, showing in Table.1. The third part is comparing the accuracy of how well our trained

facial recognition model performed on our testing images and the accuracy of images processed by Fawkes. The procedure of this step is shown in Fig.5.

The conclusion from the preliminary experiment results is that Fawkes is proven to be effective in Evasion Attack scenarios so we can proceed with our analysis.

Experiment A

The data used at this stage is derived from the VGGFace2 dataset. VGGFace2 dataset is a large-scale face image dataset that contains around 3 million images of almost 10,000 subjects. The dataset has large variations in pose, age, illumination, ethnicity and profession [10], which is why we want to leverage this for curating our input dataset of diverse individuals.

Sixty distinct individuals are extracted from the source VGGFace2 dataset and we use them as our 'Categorized Input', show in Table.2. For each individual, we have around 60-80 different images capturing varying angles, backgrounds, and lighting. After preprocessing and cleaning up the dataset, we then split this preprocessed dataset into a 75-25 train-test split where we then train a custom facial recognition model leveraging Microsoft Cognitive Services Face Client library's PersonGroup API .2. The data used for APIs is evenly distributed between the categorized inputs in Table.2 where we have the same numbers of different angles, backgrounds and lighting images for each distinct individual.

We curated this set of individuals to ensure even balance and representation of race and gender to test against Fawkes for this initial experiment because we were not able to find a large enough labeled population to randomly and confidently be able to select my dataset and ensure equal representation since the purpose of my thesis is centered around testing bias and fairness of race and gender of Fawkes performance.

As described previously, we first want to test our hypothesis of whether there exists statistically significantly different results in privacy protection performance for different races and genders under Fawkes model. The Max Iterations here denotes the times of max computation of gradient descend algorithm. Learning Rate is the speed of convergence that is used in Fawkes to get optimal perturbations.

For this experiment, we fixed the parameters 'Max Iterations' and 'Learning Rate' (Max Iterations = 200, Learning rate = 20: these values are the mid-mode for Fawkes model Table.1 and chosen to ensure sufficient Fawkes model performance) with constant values and varied DSSIM to test how DSSIM changes with different categorized inputs (DSSIM = v1, v2, v3). We ran this experiment to test our hypothesis that for different categorized input groups with the same amount of perturbations added v_i (i = 1,2,3), Fawkes's performance is statistically significantly different between the groups. For each of our categorized input group split on race (Caucasian, Asian and African) and fixed on gender (Female), we add the same amount of perturbations per image and ran them through our pre-trained image classifier from the Face Client library from Microsoft Cognitive Services API .1 to generate distributions of confidence values for each categorized input group. As we vary the DSSIM values, we take the average of confidence values generated per group to test how Fawkes performs across different racial groups. Then we run independent two-sided t-tests (assuming unequal variance) to test against the null hypothesis that the different racial groups' average performance values are the same. So we essentially have three sets of null hypothesis to test against that Fawkes is proven to be effective in Evasion Attack scenarios so we can proceed with our analysis.

As described previously, we first want to research relationships between facial properties and outputs. The first step is to prove facial properties have relationships with

the final results. To prove such relationships, we generate the data and make experiments to accept or reject our hypothesis Table.3.

Note: Interpretation of p-Value: p-value—less than alpha value means statistically significantly different.

1. μ_1: true average of confidence values of AsianFemale
2. μ_2: true average of confidence values of BlackFemale
3. μ_3: true average of confidence values of WhiteFemale

DSSIM	Groups	Hypothesis Test
0.001	Asian Female Vs African Female	$H_0 : \mu_1 = \mu_2, H_1 : \mu 1 \neq \mu_2$
	Asian Female Vs Caucasian Female	$H_0 : \mu_1 = \mu_3, H_1 : \mu 1 \neq \mu_3$
	African Female Vs Caucasian Female	$H_0 : \mu_3 = \mu_2, H_1 : \mu 3 \neq \mu_2$
0.005	Asian Female Vs African Female	$H_0 : \mu_1 = \mu_2, H_1 : \mu 1 \neq \mu_2$
	Asian Female Vs Caucasian Female	$H_0 : \mu_1 = \mu_3, H_1 : \mu 1 \neq \mu_3$
	African Female Vs Caucasian Female	$H_0 : \mu_3 = \mu_2, H_1 : \mu 3 \neq \mu_2$
0.009	Asian Female Vs African Female	$H_0 : \mu_1 = \mu_2, H_1 : \mu 1 \neq \mu_2$
	Asian Female Vs Caucasian Female	$H_0 : \mu_1 = \mu_3, H_1 : \mu 1 \neq \mu_3$
	African Female Vs Caucasian Female	$H_0 : \mu_3 = \mu_2, H_1 : \mu 3 \neq \mu_2$

TABLE 3: Hypothesis Table

In Formula 4.1 *n* is the number of images under one pair.

$$\frac{1}{n}\sum_{i=1}^{n} Classifier(Fawkes(Image_i, DSSIM, 200, 20)) \tag{4.1}$$

Table.4 describes parameter settings of Experiment A. After finishing computation, we can use the Average Confidence to plot and find new optimal thresholds corresponding to different categorized inputs based on race and gender. By setting 0% - 25%, 25% - 50%, 50% - 75%, 75% - 100% of of each confidence curve, we can find DSSIM values as new thresholds corresponding to different categorized input groups. The stratification of final results will be very important, which means that we've shown the Fawkes model

protects different images from differing races and genders at a statistically significantly different rate, where generally, we need to add more perturbations for darker skin toned individuals in order to achieve the same level of privacy protection by Fawkes.

DSSIM	Max Iterations	Learning Rate
0.001	200	20
0.005	200	20
0.009	200	20

TABLE 4: Experiment A Settings.

Experiment B - Analyzing Relationships Between Max Iterations and Learning Rate

Similar to Experiment A, but we vary different parameters, show in Table.5. We vary Max Iterations and Learning Rate while holding DSSIM value fixed as a constant value (still chosen from Fawkes model mid-mode Table.1 to ensure sufficient Fawkes model performance) in order to find the optimal Max Iterations and Learning rate for our categorized input groups in C.2. Before searching for optimal parameters, we leverage Experiment B to analyze the relationships among Max Iterations, Learning Rate and output confidence.

In Experiment B, we leveraged different permutations of the original Fawkes setting Table.1 in order to analyze the relationship that how these different permutations of Max Iteration and Learning Rate effect confidence values of our trained custom facial recognition model. Initially, we fix the DSSIM value at 0.005 (set at Fawkes's mid-mode setting in order to have enough perturbations) and vary the Max Iterations and Learning Rate as shown below in Table.5.

Learning Rate / Max Iterations	20	35	40
20	(20,20)	(20,35)	(20,40)
50	(50,20)	(50,35)	(50,40)
200	(200,20)	(200,35)	(200,40)

TABLE 5: Parameter Setting Configurations.

Experiment C - Searching for the Optimal Parameters

Experiment C modifies, expands and builds upon Experiment A and Experiment B with more input data and features (e.g., adding additional ”Male” gender and increasing the size of categorized input data), with parameter settings configured in Table.6. We also want to search for optimal parameters by applying exhaustive searching.

The categorized input settings of this experiment includes three different races - African, Asian and Caucasian adults and two genders, Male and Female. We are going to use fixed Max Iterations and Learning Rate with changing DSSIM values Table.6. As described, we will then have six different pairs of groups of people Table.6. The purpose of making these pairs is to find the new optimal DSSIM for each group, also known as the optimal amount of perturbations to add to each group by Fawkes to ensure equal privacy protection.

DSSIM	Learning Rate	Max Iterations
0.002	40	120
0.003	40	120
...	...	...
0.008	40	120

TABLE 6: Configuration for Data Generation C.1.

From Experiment B, it is clear that using large Max Iterations and large Learning Rate does not have good performance and using large Max Iterations will have huge time consumption as shown in Time Consumption Experiments 5.5. Based on prior work and

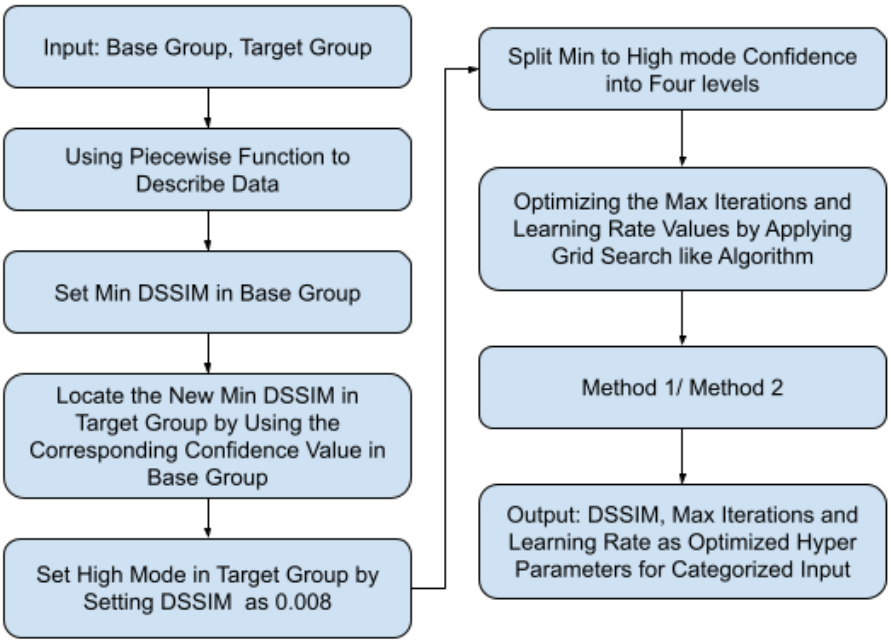


FIGURE 6: Procedure of Acquire New Parameters.

specifically analysis done in Experiment B 5.3, we set Max Iterations as 120 and Learning Rate as 40 for optimal Fawkes model performance.

We use Formula 4.2 to describe how we generate the data we are going to use. n denotes the number of the images we uses in one pair Table.2.

$$\frac{1}{n}\sum_{i=1}^{n} Classifier(Fawkes(Image_i, DSSIM, 120, 40)) \tag{4.2}$$

We leveraged the Grid Search Algorithm to search for the optimal DSSIM is: a.setting base using one of the pairs whose overall performance is the best; b. after acquiring the base case, we will set one DSSIM value for the base case as the min mode. Fig.7 describes the selection of new Min mode. Since we have the DSSIM and corresponding (average) confidence value X, by using this X, we may find the minimum DSSIM value for other groups of people. The detailed description is shown in Algo.1.

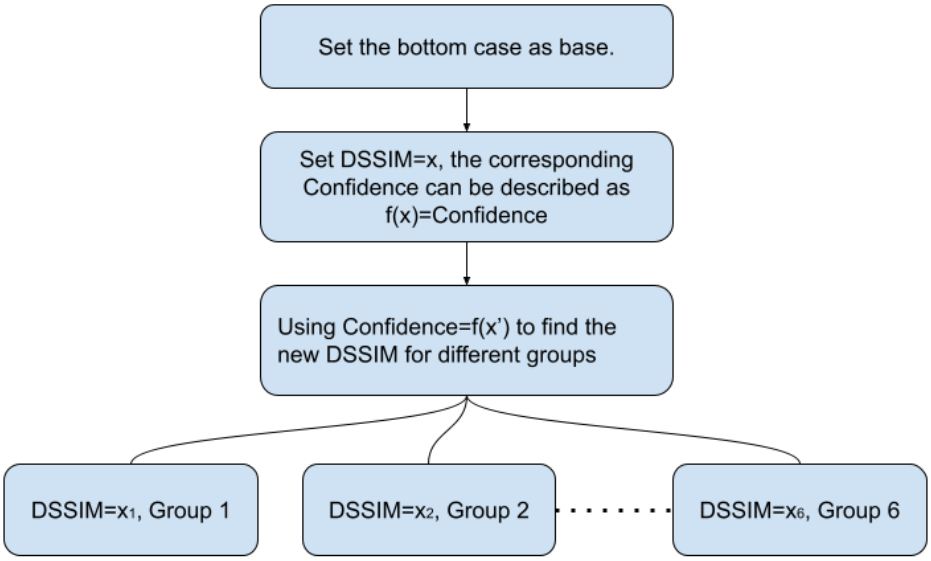


FIGURE 7: Find Optimal DSSIM.

Algorithm 1 New Min DSSIM

1: **procedure** NEWMINDSSIM($BaseGroup$, $BaseMin$, $TargetGroup$)
2: $BaseGroupF$ = Convert_DSSIM_Confidence($BaseGroup$)
3: $TargetGroupF$ = *Convert_Confidence_DSSIM*($TargetGroup$)
4: $NewMin$ = $TargetGroupF(BaseMin)$
5: **return** $NewMin$
6: **end procedure**

Note:

Convert_DSSIM_Confidence(): Convert Data to a Piece-wise Function. DSSIM as variable, Confidence as results

Convert_Confidence_DSSIM(): Convert Data to a Piece-wise Function. Confidence as variable, DSSIM as results

We leveraged different DSSIM together with fixed Max Iterations and Learning Rates determined by Experiment B analysis where we found that large Max Iterations (like 200) does not tend to have good performance when having large Learning Rate. We

decide to use 120 Max Iterations and 40 Learning Rate in C.1. We leveraged input pairs based on Table.2 and applied settings based on Table.6.

C.2 uses the newly-acquired DSSIM values from C.1. Max Iterations and Learning Rate are not the main threshold that control the effectiveness and amount of perturbations of algorithm, but it helps with finding most optimized data points to make confidences values as low as possible. From previous sections, the pattern of Max Iterations and Learning Rate are different for different groups of people. If we fix various DSSIM values, we want to find optimal Max Iteration and Learning Rate values based on minimizing resulting model confidence. There are two main ways to do this that are differentiated by amount of compute time, where one method takes much longer than the other–with the same results in Max Iteration and Learning Rate, configurations in Table.6.

In C.1 we acquired ideal DSSIM values for different groups of categorized inputs. In this section, the Max Iterations and Learning Rate values are tuned to generate optimal Fawkes model performance results. In the following, we will describe in detail the two main methodologies designed to find optimal Max Iteration and Learning rate values.

The working flow of the first methodology is shown in Fig.8.

In order to acquire ideal Max Iterations and Learning Rate values, we design two methods. Both Method 1 and Method 2 will have the same results, the difference is that when doing Method 1, the user can get the filtered records each round. The reason of making this is that we considered there exists some special scenarios. For example, in the last several steps, we found that there exists two records with similar confidence values, but the Max Iterations values are different. As described in time consumption part, Max Iterations is the key parameter that costs huge time. By providing these records, users can choose the relative good result with smaller Max Iterations value in order to save Fawkes processing time.

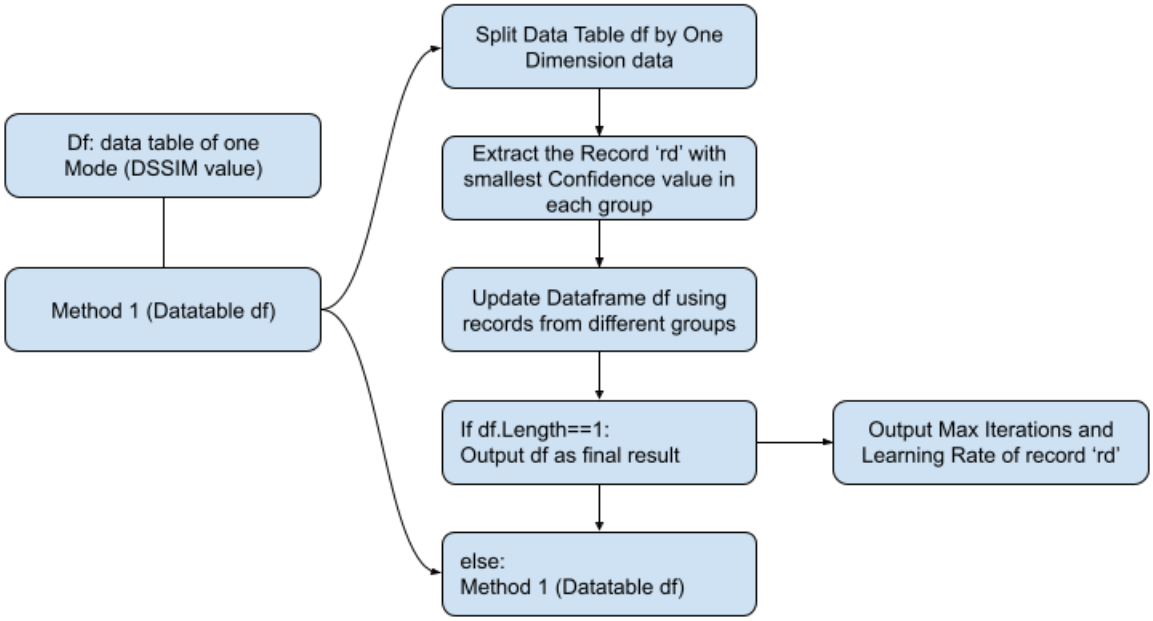


FIGURE 8: Method 1 Working Flow.

Method 1

For the first methodology, we consider compute time and aim to minimize compute time as well as provide users with a selection of optimal Max Iteration values to choose from through displaying a step-by-step guide of optimization, shown in figures in appendix .5. Each time, we change one parameter and fix other parameter. For example, if we are going to set Max Iterations as the varying parameter, we will fix DSSIM values and Learning Rate values. We aim to reduce the influences from other dimensions and improve the efficiency as much as possible.Please refer to Algo.2 below for detailed implementation.

This method shows results step by step. More details and examples are shown in Appendix .5. We found that Max Iterations heavily influences the total run time 5.5. The setting for Max Iterations is 40, 80 and 120. The reason of choosing these three numbers originates from Preliminary Experiment, Experiment A and Experiment B. C.1 which describes the relationships of Max Iterations and Learning Rate with final results. As

Algorithm 2 Method 1

procedure METHOD1($Datatable\ df$)
    Initialize New Data Table New_df ▷ Store new records
    Data tables df group by One Parameter A ▷ df contains data tables
    for $\_df$ `in` df **do**
        $record\_index$=$\_df[\_df.Conf = \_df.Conf.min()].index$
        New_df.append($\_df[\_df.index == record\_index]$
    end for
    if $len(New\_df) == 1$ **then**
        return New_df
    else
        Method 1(New_df)
    end if
end procedure

stated in C.1, large Max Iterations values sometimes will not improve the performance much, so we decide to have the largest Max Iterations set as 120. The setting for Learning Rate values is 10,20,30,40. There are two reasons of setting these numbers: 1) Data from Fawkes source code settings on Learning Rate; 2) Expanding the range based on C.1, making Max Iterations and Learning Rate as our searching targets for the optimal parameters.

Method 2

This method does not take into account time consumption and picks out the optimal pair of Max Iteration and Learning Rate directly by taking the Max Iteration corresponding to the minimum confidence values. This method saves much time, but it will not give out steps and comparison in each stage.

Algorithm 3 Method 2

procedure METHOD 2($Datatable\ df$)
    $record\_index$=$df[df.Conf = df.Conf.min()].index$
    return $record$
end procedure

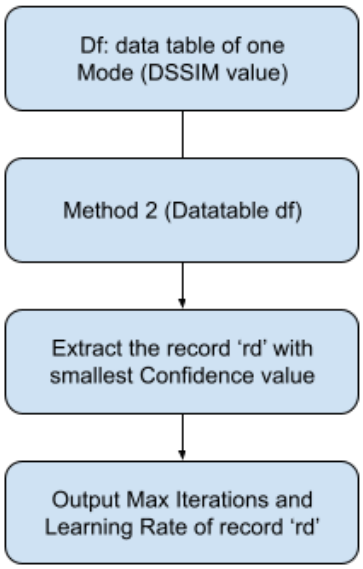


FIGURE 9: Method 2 Working Flow.

For more on time consumption and cost analysis, please refer to ”Time Consumption Experiments” 5.5 in ’Results’.

CHAPTER V

RESULTS AND DISCUSSION

Preliminary Experiment - Feasibility in Evasion Attack

In this experiment, we had three different kinds of input data: 1) original testing data (the images without perturbations); 2) Processed Data - Mid Mode, using Fawkes to process original testing data by applying Mid-mode settings perturbations; 3) Processed Data - High Mode, using Fawkes to process original testing data by applying High mode. Different mode information can be find in Table.1. After we trained a facial recognition model on our training set post 60-40 train-test split, we applied the trained model on the test set where we then took the average of all the confidence values from the test set– which came out to be around 81% for the original test images, 75% for the mid-mode processed images and 30% for high mode processed images. From the testing results shown in Fig.10-11-12, the confidence of recognition system keeps decreasing when applying robust modes (Mid Mode and High Mode) of Fawkes.

The results are described by using histograms and kernel density estimation (KDE) plots (Fig.10-11-12) to visually plot out the distribution of confidence values for each of the three inputs of images. Histograms here are used to described the frequency of confidence, and we also make KDE plot to show the shift of confidence values when applying much perturbation in Fig.10-11-12.

Since the main threshold (from introduction to Fawkes, Section.II) that controls the effectiveness of perturbations is DSSIM values, based on the results we have above, we can say that as we increase DSSIM values, the average confidence of our trained

facial recognition model will decrease. In conclusion, Fawkes is shown to be effective in Evasion Attack scenario.

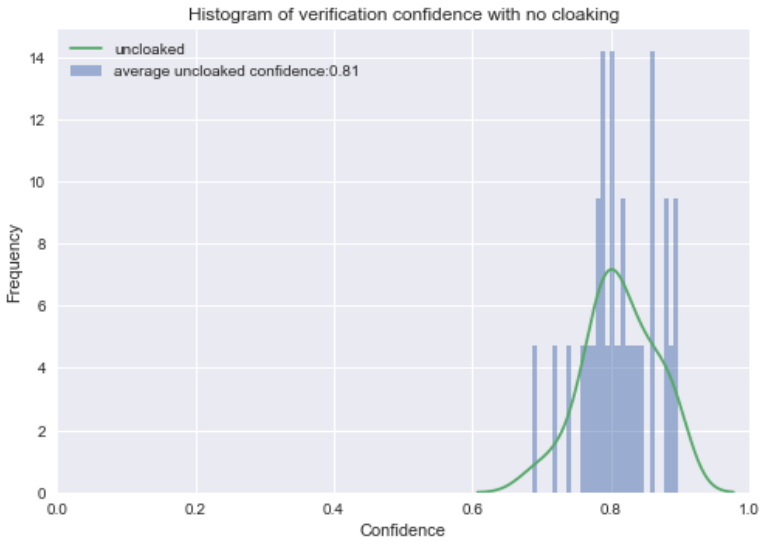


FIGURE 10: Original Test Images, No Perturbations.

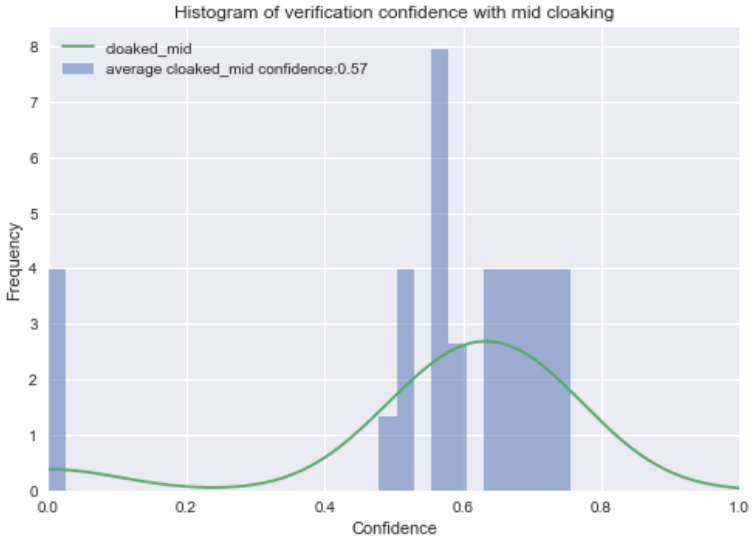


FIGURE 11: Processed Images - Mid Mode(DSSIM = 0.005).

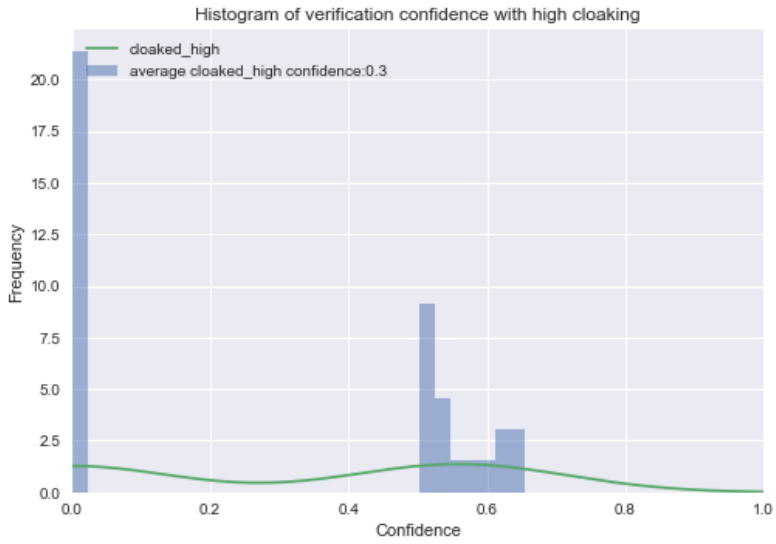


FIGURE 12: Processed Images - High Mode(DSSIM = 0.008).

Experiment A

The results of experiment is shown below in Table.4 with hypothesis testing results to test for statistically significant difference between the categorized input groups protection performance when inputted into Fawkes model in Table..6 ”n00xxxx” is the name of that class that denotes racial groups, in this case, Asian, African or Caucasian. Experiment A is built to prove our assumption in Experiment A 4.2 in Experiment Section. We use three different DSSIM values with the same Max Iterations and Learning Rate values. The reason is 1) Three DSSIM values used here vary a lot, it can provides the overview though it cannot provide with accurate data; 2) 200 Max Iterations and 20 Learning Rate are the same as origin Fawkes’s Mid mode settings, settings shown in Table.4.

The configurations of the input data are shown in Table.4. The data described in Table.8 are plotted as Fig.13 describes the test results from our categorized input of the different groups of individuals based on race after applying Fig.4. Confidence

DSSIM	Groups	p-Value (alpha = 0.05)
0.001	Asian Female Vs African Female	0.0432
	Asian Female Vs Caucasian Female	0.00561
	African Female Vs Caucasian Female	0.000348
0.005	Asian Female Vs African Female	0.0436
	Asian Female Vs Caucasian Female	0.270
	African Female Vs Caucasian Female	1.59e-05
0.009	Asian Female Vs African Female	0.0193
	Asian Female Vs Caucasian Female	N/A
	African Female Vs Caucasian Female	0.0193

TABLE 7: Hypothesis p-Value Table

DSSIM	AsianFemale	AfricanFemale	CaucasianFemale
0.001	0.569	0.703	0.499
0.005	0.173	0.438	0.025
0.009	0	0.138	0

TABLE 8: Experiment A Test Results

values decrease as DSSIM increases across all three categories of input, which means DSSIM values are the main constraint for Fawkes system. The effectiveness of lowering confidence for three groups are different. As shown in Fig.13, for the same DSSIM thresholds, the average final results(average confidence) are different. After running our hypothesis test (Table.7) to test for whether the privacy performance is statistically significantly different, we get the following results in Table..6 for an alpha level of 0.05.

The results of the experiment is shown below in Table.10. ”n00xxxx” is the name of that class or that person. From Table..6, we see that after running two-sided independent t-tests assuming unequal variance between the confidence values of the paired groups (e.g., Asian Female Vs African Female) at DSSIM = 0.001, 0.005, 0.009–we see that all but one relationship, Asian Female Vs Caucasian Female, be statistically significantly different. This implies that the amount of perturbations that need to be added in order for images of different races (holding gender the same) is statistically significantly different

for Caucasians, African and Asians. This proves our assumption in Fawkes model treating different races differently in terms of privacy protection effectiveness.

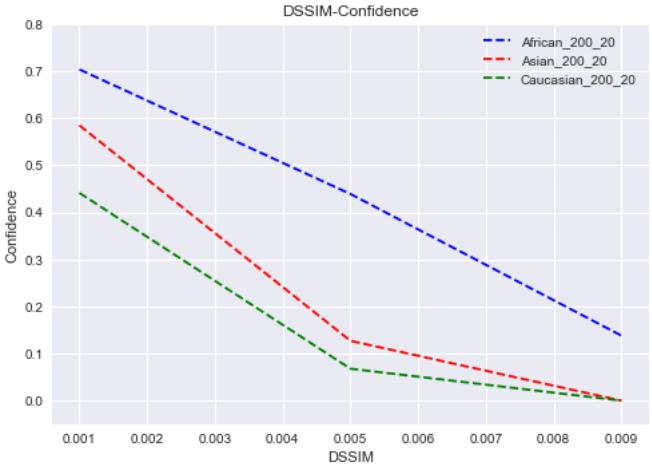


FIGURE 13: Test Result Plot.

Experiment B - Relationship between Max Iterations and Learning Rate

2-D Plots Analysis

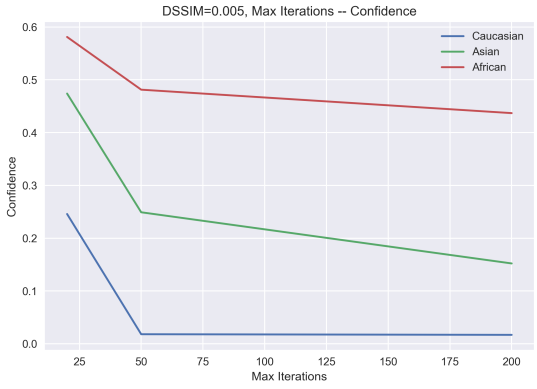


FIGURE 14: View From Max Iterations

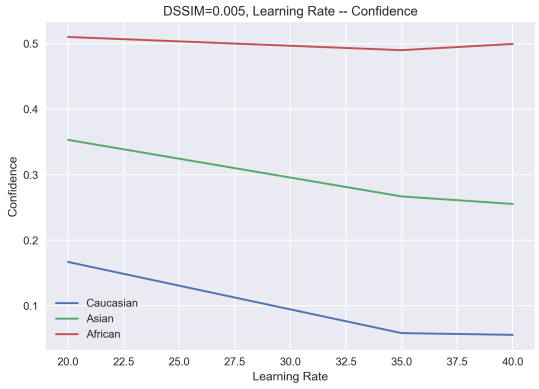


FIGURE 15: View From Learning Rate

The first plot Fig.14 shows the relationship between Max Iterations and Confidence under DSSIM = 0.005. Specifically, all the values used here are calculated through

averaging all the confidence values generated per image from the trained facial recognition model. It means the confidence values of the same item which has the same Max Iteration value and different Learning Rate values will be averaged. The first plot shows the averaged confidence from Max Iteration angle. We draw two conclusions as discussed below.

1. Three confidence values of groups of people have stratification, which means the performance of these groups varies a lot. From Result 1 (Fig.14), we can see the difference of confidence values of these three groups.
2. The influence of Max Iteration for three different groups of people is quite different. As shown in the subplot, small Max Iterations (20) has same affect two three groups. Medium Max Iterations can lower confidence values for Asian and Caucasian.
3. Max Iterations affects Caucasian heavily. Max iteration also influence Asian a lot, the confidence of Asian decreasing fast when increasing the value of Max Iteration. For African, the Max Iteration also works, but comparing with the two other groups of people, its performance is the weakest.

The second plot Fig.15 shows the averaged confidence from Learning Rate angle. From this plot, we could have such conclusions.

1. The second plot is showing the relationship between Learning Rate and Confidence. The stratification still exists in this plot.
2. Increasing learning rate in order to lower the confidence works for Asian and Caucasian. But for African, it fails to reach the intentions.

From these two plots using average values, we can draw conclusion that increasing the value of Max Iterations will lower the confidence for three groups of people.

Increasing the value of learning rate does not work well for African group, which means increasing cannot lower the confidence values on average, though it does affect Asian and Caucasian.

3-D Graph Analysis

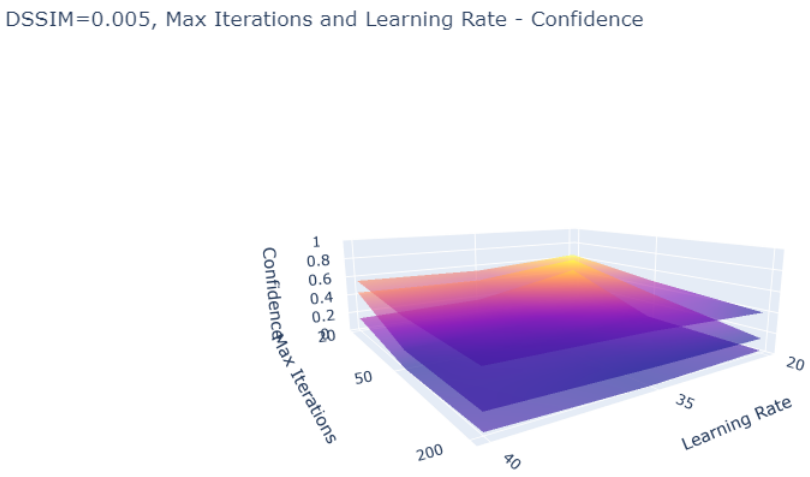


FIGURE 16: 3-D Result Plot.

Fig.16 is showing the overall tendency when Max Iterations and Learning Rate works together on classification confidence.

There are three layers in this plot. The upmost layer denotes African. The middle layer denotes Caucasian. The bottom layer denotes Caucasian. From these layers, we can see the stratification clearly, this implies that for all different images in the categorized input groups, although they have the same parameter settings (Max Iteration and Learning Rate), the performance of Fawkes in protecting the privacy of the images is different.

In upmost layer, it has one highest confidence point and one lowest confidence point. The highest confidence is when Learning Rate = 20 and Max Iterations = 20. The lowest confidence is when Learning Rate = 20 and Max Iterations = 200.

In the middle layer, it has one highest confidence point and one lowest confidence point. The highest confidence is when Learning Rate = 20 and Max Iterations = 20. The lowest confidence is when Learning Rate = 20 and Max Iterations = 200.

In the bottom layer, it has one highest confidence point and two lowest confidence points. The highest confidence is when Learning Rate = 20 and Max Iterations = 20. The lowest confidence is when Learning Rate = 35 and Max Iterations = 200 or Learning Rate = 40 and Max Iterations = 50.

Subplots Analysis

The subplots in Fig.17 show the behaviors of confidence when fixed Learning Rate value or Max Iterations value. This subplots present the same groups of data with 3-D plots. The subplots shown are the cross sections of 3-D plot. Subplot (1-1), Subplot (1-2), Subplot (2-1) show the tendency of Confidence when increasing the value of Max Iterations with different fixed Learning Rate values. For three groups of categorized input, the tendency of confidence values behaves similarly, showing us that by applying large Max Iteration while also increasing Learning Rate, the average confidence of the trained facial recognition model can be lowered (meaning better privacy protection for the images). However, when setting large Max Iterations and Learning Rate at the same time, the results, average confidence, tend to be unstable. Thus, we cannot set large Max Iterations and Learning Rate values at the same time.

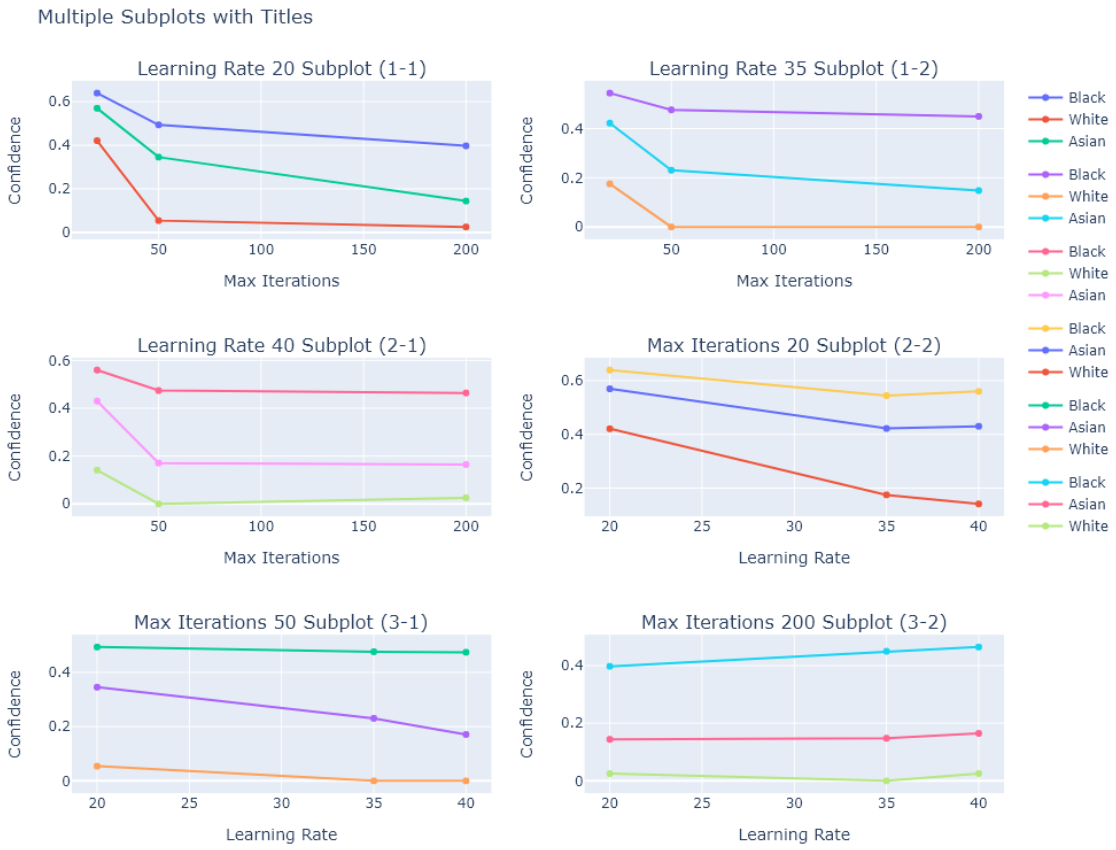


FIGURE 17: View with Subplots

Subplot (2-2), Subplot (3-1), Subplot (3-2) show the tendency of Confidence when increasing the value of Learning Rate with different fixed Max Iterations values. The tendency of confidence values behaves differently when fixed the value of Max Iterations.

In Subplot (2-2) and Subplot (3-1), for Asian and Caucasian, the confidence values keep decreasing when increasing learning rate with small Max Iterations values (20 and 50). For all three groups, confidence value has an overall decreasing tendency when Max Iterations is 20 and 50. When setting Max Iterations as 200, the tendency of confidence is quite different from previous plots. For three different groups of people, the tendency is overall increasing, which means increasing Max Iterations (200) and learning rate (40) at

the same time is not a good direction for lowering classification confidence. Large Max Iterations and Large Learning Rate cannot help with solving optimization problem ideally.

As shown in subplots, using larger Max Iterations and smaller Learning Rate will have good results . In most cases, Max Iterations has large influence on final results because it denotes the maximum computation times. Larger Max Iterations denotes longer processing time. In most cases, the more iterations are, the better the results will be. However, when combing the subplots and 3-D plots, we can have another conclusion that tuning the learning rate and max iterations in some cases can achieve the same results as using larger Max Iterations and small learning rate. Sometimes it may avoid confidence ”rising up”. This means if we can control the relationships of Max Iterations and Learning Rate properly, we can use relatively small Max Iterations and proper Learning Rate to achieve the same results as larger Max Iterations and small learning rate. Using relatively small Max Iterations can lower the processing time.

For example, Subplot (1-1), Subplot (2-1) and Subplot (3-1) the curve of Caucasian shows that big learning rate and small max iterations can achieve the same results as large max iterations. And in Subplot (3-2), large max iterations and large learning rate may lead to confidence ”rising up”, the procedure fail to find the local minimum in the searching process. From Experiment B, we better understand the dependent relationship between these Fawkes' parameters, Max Iteration and Learning Rate.

Experiment C - Searching for the Optimal Parameters

C.1 uses different DSSIM values together with fixed Max Iterations and Learning Rate values. C.1 includes two genders and three skin tones, which means that it contains 6 gender-skin-tones pairs as show in Table.9. For each pair, five different people will be

extracted randomly from the 60 people dataset that comes from VGGFace2 dataset [10]. Basically, C.1 is the enhanced version of Experiment A made in Proposal stage.

The computation flow is: a. Using Fawkes generate different pairs of data; b. Using APIs to compute the confidence with trained APIs, the concrete procedure is depicted in Fig.4.

The results of confidence data after applying Fig.4 is shown in the plot 18. The generated data is acquired through calculating the average confidence of each DSSIM point. The detailed data description is shown in the Appendix. The summary of data is shown in Table.9.

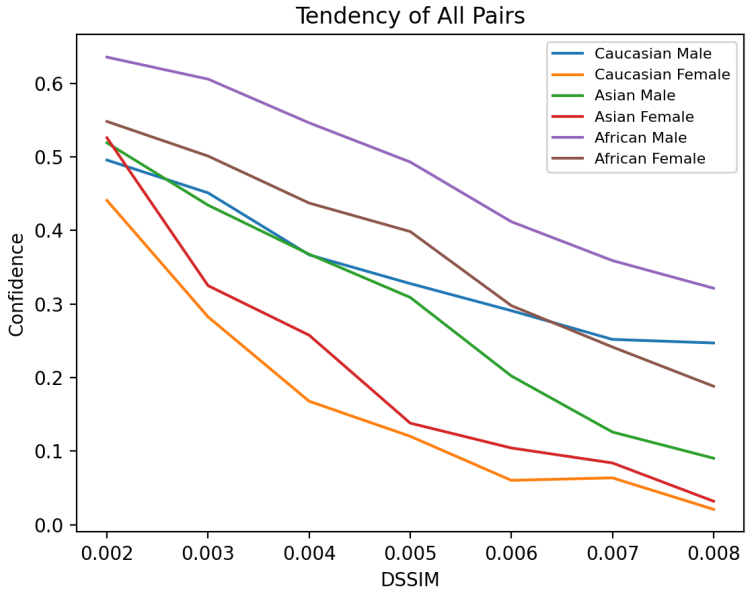


FIGURE 18: Average Confidence of All Groups - DSSIM.

Table.9 describes output confidence of the data processed by using DSSIM from 0.002 to 0.008 with 120 Max Iterations and 40 Learning Rate. The average confidence of using different DSSIM values together with fixed Max Iterations and Learning Rate can be visualized from two perspectives, genders Fig.19 and skin tones Fig.20. As shown in the Table.9, from when we hold race constant and only look at average confidence per

Record	Pair	Average Confidence	Standard Deviation
1	Caucasian Female	0.16524	0.26284
2	Caucasian Male	0.3539	0.32667
3	Asian Female	0.20973	0.28059
4	Asian Male	0.29421	0.30194
5	African Female	0.35578	0.30691
6	African Male	0.48127	0.28996

TABLE 9: C.1 Results Summary.

gender, the confidence of Females are relatively smaller than the Males when it comes to the facial recognition model accurately detecting the images meaning the protection for females are better than males. While when we hold gender constant and only look at average confidence per race, the protection for Caucasian and Asian is much better than African groups (Fig.19 and 20, respectively).

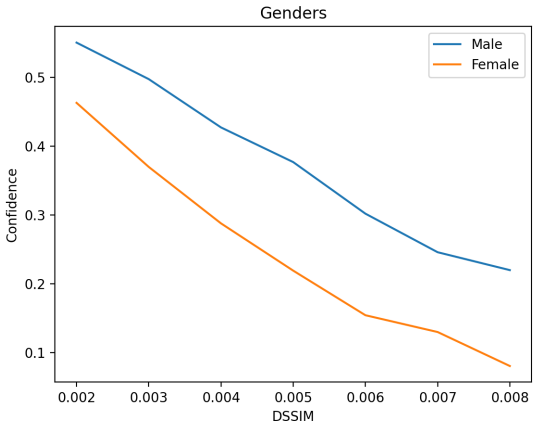


FIGURE 19: Genders.

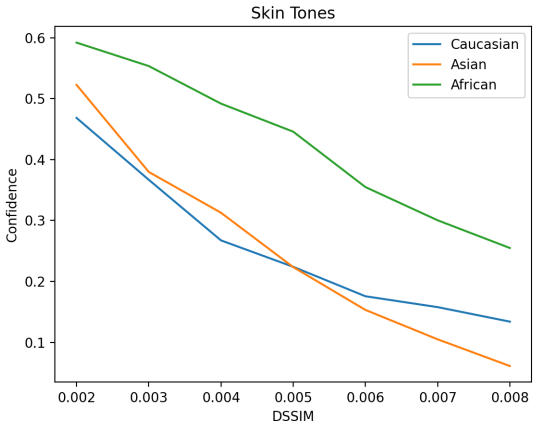


FIGURE 20: Skin Tones.

The data in Fig.18 is computed via averaging all the confidence values of individual images after fitting the pre-trained facial recognition model. Each curve denote one specific race-gender pair. In Fig.18, it is obvious that the data of different pairs stratifies much. The bottom curve is Caucasian-Female adult group, the uppermost curve is African-Male adult group. By increasing the DSSIM values, we can see that the

performance are different even when these groups have same DSSIM, Max Iterations and Learning Rate values. Faced with this inequality, setting large DSSIM values will be the most powerful approach to mitigate this bias.

By increasing the DSSIM values, we can see that although the confidence will be lowered by setting big DSSIM values which means add stronger perturbations, it does treat different groups with inequality. Faced with this inequality of weak protection to some specific groups, setting larger DSSIM values will be our primary approach.

In Fig.18, we observe that under fixed 120 Max Iterations and 40 Learning Rate, Caucasian groups has the overall best performance, Asian groups rank the second and the African groups rank the third. Our basic solution is to split six groups with two genders and use the best case of that gender groups, we called base case, to find the best optimal to those whose performance is not as good as the base case.

The procedure can be described briefly as: a.Split pairs into two genders, Female groups and Male groups; b.Set base case; c.Find corresponding confidence value in target curve; d.Locate the target DSSIM value in target curve. The step b-d is the summary of Algo.1. The newly found DSSIM is the new Min mode.

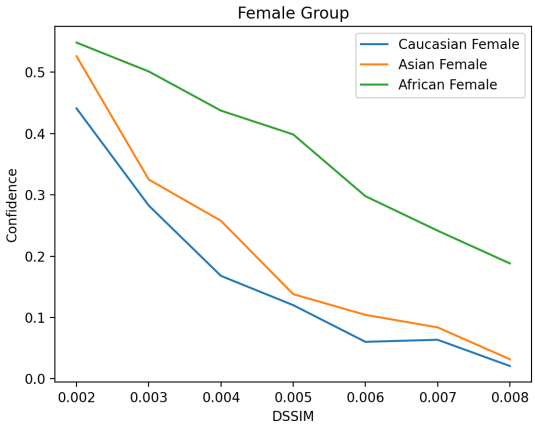


FIGURE 21: Female Group

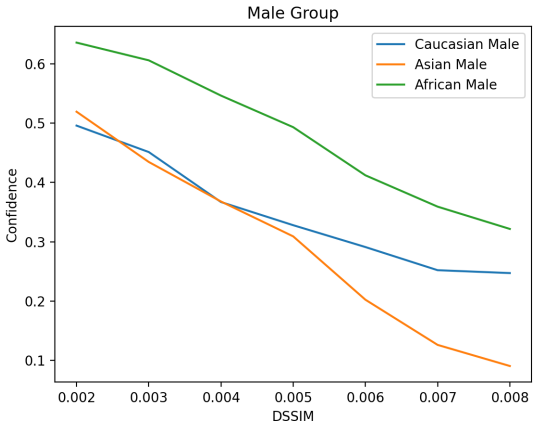


FIGURE 22: Male Group

More detailed description about the steps above:

1. Split the pairs into two groups: Female group and Male group.
2. Set base case(curve) which is one pair of six pairs listed above. Then set DSSIM = x as min mode, the corresponding Confidence value is y. In target curve, use y to locate the new DSSIM value.
3. Find optimal parameters using two groups, Females and Males. The procedure can be described as: we set DSSIM = X as min for base case, the corresponding average confidence is Y. Then we use the Y to locate the DSSIM value X' in target case. The X' here are function as the lower bound, the higher bound is going to be DSSIM = 0.008 . Since with large DSSIM value, the perturbation will be heavier.

Female Group

As described Algo.1 and descriptions above, we first set Female Caucasian pair as base case. In the base case, we set min mode as base case's minimum DSSIM value (0.0024). Then we use the corresponding confidence to locate the new DSSIM for Asian Female as new Asian Female minimum mode DSSIM value. After we acquire the Asian Female Minimum mode, we then use Confidence (DSSIM = 0.0024) and Confidence (DSSIM = 0.0080) as higher bound and lower bound. By using these two bound, we apply similar method to acquire the new DSSIM. They are Min (DSSIM = 0.0024), Low (DSSIM = 0.0033), Mid (DSSIM = 0.0047) and High (DSSIM = 0.008).

The newly acquired DSSIM are also tested and compared with origin min 0.002 (Fig.23). In Fig.23, "before" denotes values in searching procedure; "test" denotes the values when applying the newly acquired DSSIMs; "Origin" denotes the values applying DSSIM = (0.002, 0.003, 0.005, 0.008) , which is the same as Fawkes original modes.

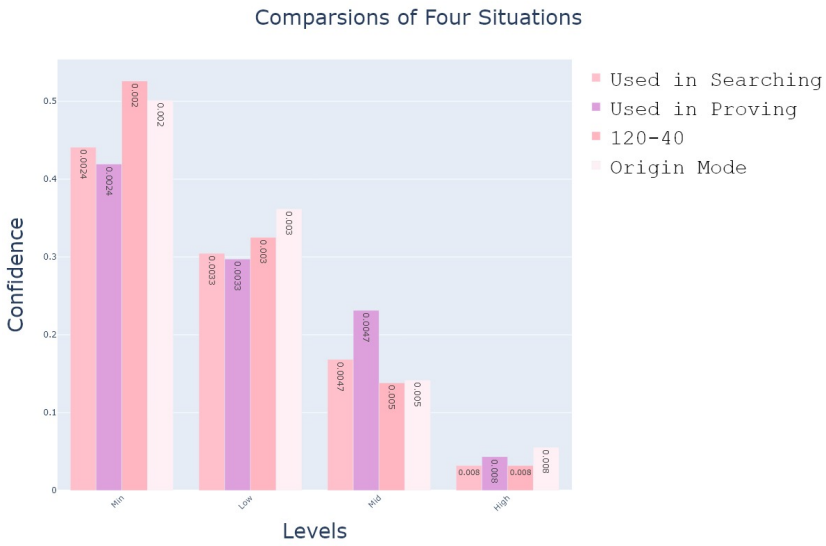


FIGURE 23: Asian Female Comparison.

When dealing with Female Africans, we applied the same strategy. We first use Caucasian Female as base, the set DSSIM = 0.002 as Min Mode for Caucasian. Then we find the optimal DSSIM values for African Female group. The new DSSIM for four modes of Asian Female are (0.0039, 0.0054, 0.0065, 0.008).

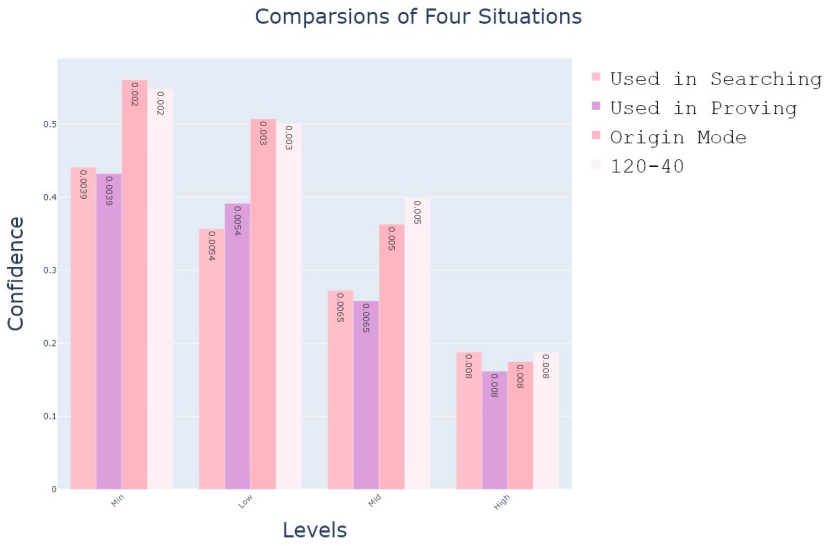


FIGURE 24: African Female Comparison.

Male Group

For Male group, there is one special case happens. We notice that two groups have overlapping and intersection data points. Since one of these two curve, Asian curve, is considered as the base according to statements above. So we won't do the procedures on Caucasian Male group and consider this as special cases. We applied the optimal parameter searching on African Male group. After the procedure, we have the bar plot below. The new DSSIM for four modes of African Male is (0.0045, 0.0055, 0.0065, 0.008). The comparison data is plotted shown in Fig.25. From the bar chart, the newly acquire DSSIM values achieve better results in Min, Low and Mid mode.

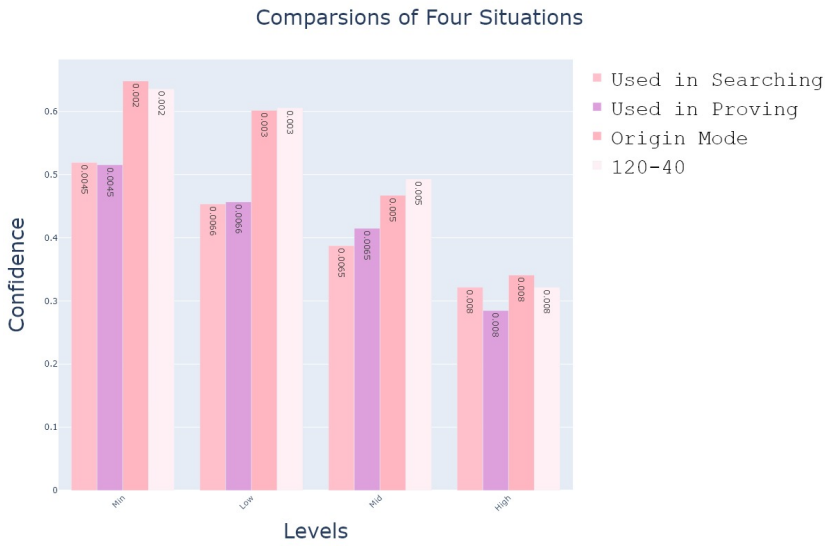


FIGURE 25: African Male Comparison.

Experiment C.2 – Evaluation using Different Max Iterations and Learning Rate

Since the experiments share the same procedures, one example will be shown in this section. The data of all experiments results is put in the appendix table of Method 1 Results and Method 2 Results in .5.

Example of Method 1

We present one example when the input is DSSIM = 0.0039 (using 39 to denote in Fig.26).The light-green colored row denotes records with the lowest confidence value. The method we take is using recursion and we present the results at each stage. The tables on the left are the results of first round, and they describe the fixing Max Iterations and making Learning Rate parameters. The left figures describe the last two rounds of Method 1. The Fig.26 shows the procedure of Method 1. The order of this example procedure is from left to right.

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	39	40	10	0.599017
1	39	40	20	0.479899
2	39	40	30	0.436109
3	39	40	40	0.429007

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
4	39	80	10	0.507677
5	39	80	20	0.455911
6	39	80	30	0.438742
7	39	80	40	0.425394

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
8	39	120	10	0.446948
9	39	120	20	0.453926
10	39	120	30	0.463591
11	39	120	40	0.456883

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	39	40	40	0.429007
1	39	80	40	0.425394
2	39	120	10	0.446948

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	39	80	40	0.425394

FIGURE 26: Method 1 Example.

Example of Method 2

This method has the same settings as Method 1. The difference is that Method 1 will give out the details in each round and Method 2 will only output the final results directly based on confidence values Algo.3. In Method 1, we acquire the new DSSIM values, this step we will optimize the new Max Iterations and Learning Rate values. The input of this method is a batch of images of a group of people of one configuration

(DSSIM, Max Iteration, Learning Rate settings). For example, in Method 1, we have New DSSIM values 0.0039. Method 2 is going to acquire the Max Iterations and Learning Rate values by taking the configuration with the lowest confidence value. By using this method we can acquire optimal Max Iterations and Learning Rate.

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	39	40	10	0.599017
1	39	40	20	0.479899
2	39	40	30	0.436109
3	39	40	40	0.429007
4	39	80	10	0.507677
5	39	80	20	0.455911
6	39	80	30	0.438742
7	39	80	40	0.425394
8	39	120	10	0.446948
9	39	120	20	0.453926
10	39	120	30	0.463591
11	39	120	40	0.456883

FIGURE 27: Method 2 Example.

Time Consumption

In this experiment, we analyze the time consumption of the Fawkes system. We used Fawkes original modes 1 conclude the time consumption as shown in Fig.28-29. Fig.28-29 denotes time consumption per image and total time consumption of 10 images respectively.

From the time consumption over Fawkes original modes in Fig.28-29, we can draw a conclusion that adding robust perturbations will cost huge time. However, from Table.1, since different modes using completely different Max Iterations and Learning Rate values, we cannot tell which parameters play key roles of make huge time consumption. Due to this we also make another two experiments about time consumption, the first one is looking into the relationships between time consumption and DSSIM values; the

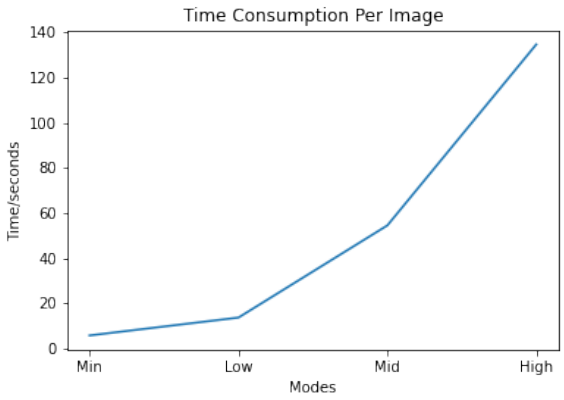


FIGURE 28: One Image

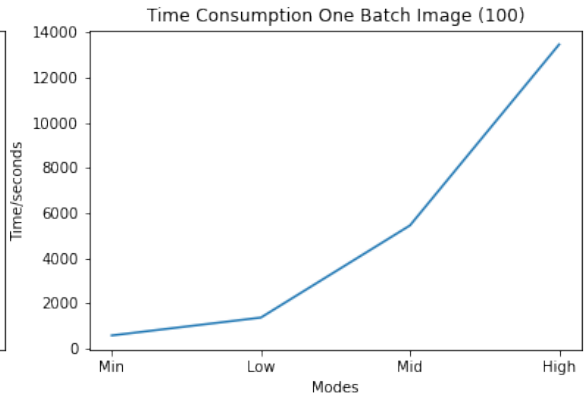


FIGURE 29: Ten Images

second will looking into the relationships with other two parameters, Max Iterations and Learning Rate.

Time consumption experiment with DSSIM. The time consumption under different DSSIM values that we take DSSIM = 0.001 to DSSIM = 0.008 and using different three different pair of settings, Max Iterations as 80 and Learning Rate as 20, 30, 40. From Fig.30 and Fig.31, the average time consumption are similar. From the boxplot Fig.30 and curve Fig.31, we can tell that the learning rate may have subtle relationships with time consumption.

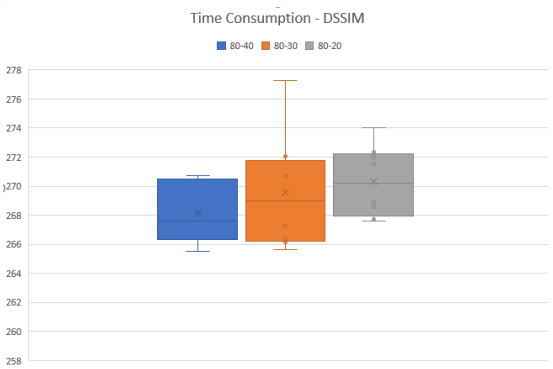


FIGURE 30: DSSIM - Time Consumption

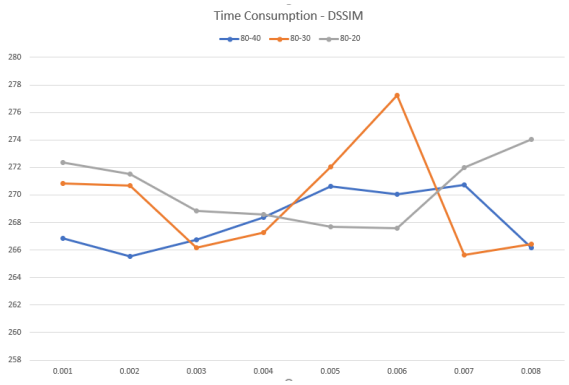


FIGURE 31: DSSIM - Time Consumption

From the result of DSSIM and Time Consumption experiment plot Fig.30. There is no relationship between DSSIM and time consumption.

Time consumption experiment with Max Iterations and Learning Rate. We apply 4 different Learning Rate values with changing Max Iterations values.

From the result of Max Iterations, Learning Rate and Time Consumption experiment plot Fig.32, Learning Rate values do not have relationships with time consumption. No matter how we change the Learning Rate values, the time consumption remains similar. Max Iterations have strong relationships with time consumption. As shown in Fig.32, the relationship between Max Iterations and time consumption is linear.

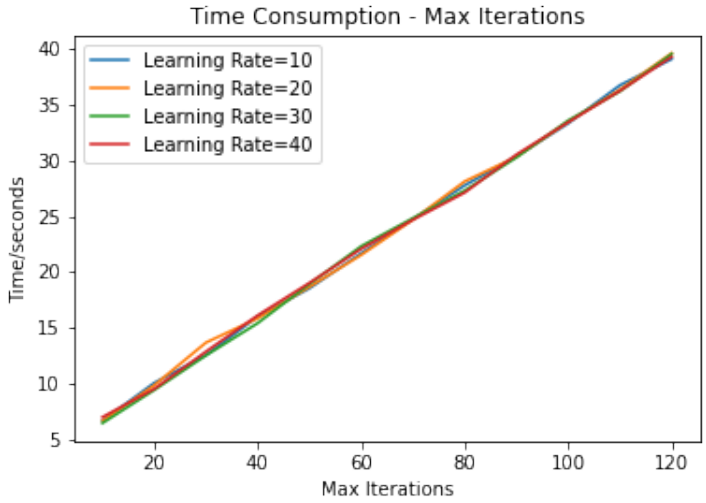


FIGURE 32: Max Iterations, Learning Rate - Time Consumption

As a conclusion of this section, the parameter Max Iterations influence time consumption heavily. The other two parameters; DSSIM and Learning Rate affect time consumption very little Fig.30.

Discussion

Our main result is a novel approach for optimal categorized input setting in the context of adversarial attack systems. We aimed to prove that the current Fawkes model protects different images from different categorized groups at different privacy protection rates (showing bias in the current model if we only use default settings for all images) and provide a set of optimal parameters for different categorized input images based on race and gender groups specified. We are essentially enhancing the current Fawkes model by taking facial properties into consideration like genders and skin tones.

Next we discuss some potential areas of improvement and enhancement to the current approach:

1. Our current model is trained on relatively small subset of hand selected individuals from different racial and gender backgrounds. This leaves room for bias in how well the trained model will perform on future test sets, thus affecting privacy protection rates for Fawkes generated outputs. We could devise a method to randomly select a larger and more racially and gender diverse group of inputs to train our facial recognition model on to further fine tune the perturbations needed from Fawkes model to ensure optimal privacy protection. We can also expand beyond race and gender and add in more demographic features like age group.
2. Sweep through more hyperparameter values for Max Iteration and Learning rate besides the fixed ones we evaluated. For example, track how Fawkes performs varying Max Iteration and Learning rate through a continuous range of values rather than the current discrete fixed values.

Some challenges faced in this thesis and initial approach are: it was extremely hard to find one dataset that can describe all the information of people (e.g., race, gender, etc.).

Also, since the data are personal images and personal description data, there are privacy protection laws surrounding accessing them.

The second idea is using continuous ranges to replace the fixed discrete values. The algorithm should ideally search within ranges instead of discrete values. The potential solution we considered is to apply differential evolution. Differential evolution can be applied here to acquire accurate Max Iterations and Learning Rate values. The aim of the thesis is tuning the hyperparameters to acquire the lowest average confidence values. By using Differential evolution, it actually searches hyper parameters within ranges, thus increasing the search ranges of hyperparameters. For example, in this thesis, we applied grid search like solution for Max Iterations and Learning Rate. As stated in the experiments, we use lists of values for our grid search. However, the intervals are too big between the discrete values for the parameters. For example, we use 40, 80, and 120 as candidate parameters. There are many numbers within the intervals that we did not take into consideration, which means we may miss the lowest confidence point. Of course, we can add more candidate values from intervals to search with more values, however, this brute force like method will be costly because it will calculate each existing pair of parameters. By applying differential evolution, the search time for the lowest point will be reduced. The metric to be applied in differential evolution version optimization will also be the average confidence values from a trained facial recognition model.

CHAPTER VI

CONCLUSION

In this thesis, we aim to show the existing bias of the current Fawkes system in image privacy protection and propose a methodology to find optimal perturbations to lower bias due to demographic features like race and gender through proposing making categorized inputs for Fawkes in Evasion Attack scenario. Our initial findings show that although the current Fawkes model have bias towards race and gender, shown by the different image privacy protection rates for images from different racial and gender backgrounds, we can fine tune the Fawkes model parameters by using our proposed methodology to ensure equal privacy protection of images by Fawkes. Fawkes's protection of different groups of people can be enhanced by applying the newly discovered method and optimal parameters of different groups of users, overall decreasing bias and ensuring a more fair model for future consideration and use.

REFERENCES CITED

[1] Muhammad Junaid Khalid, "Grid search optimization algorithm in python." https://stackabuse.com/grid-search-optimization-algorithm-in-python/.

[2] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, "Fawkes: Protecting personal privacy against unauthorized deep learning models," in *Proc. of USENIX Security*, 2020.

[3] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[4] Patrick Farley, Wade Pickett, Kraig Brockschmidt, Matthew Sebolt, "Quickstart: Use the face client library." https://docs.microsoft.com/en-us/azure/cognitive-services/face/quickstarts/client-libraries?pivots=programming-language-python&tabs=visual-studio, 2020.

[5] Alex Najibi, "Racial discrimination in face recognition technology." https://sitn.hms.harvard.edu/flash/2020/racial-discrimination-in-face-recognition-technology/, 2020.

[6] Wikipedia contributors, "Principal component analysis — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Principal_component_analysis&oldid=1023655415, 2021. [Online; accessed 21-May-2021].

[7] Microsoft, "Quickstart: Use the face client library." https://docs.microsoft.com/en-us/azure/cognitive-services/face/quickstarts/client-libraries?tabs=visual-studio&pivots=programming-language-python, 2021.

[8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[9] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[10] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pp. 67–74, IEEE, 2018.

[11] Charles Kapelke, "Adversarial machine learning." https://medium.com/cltc-bulletin/adversarial-machine-learning-43b6de6aafdb, 2019.

[12] Wikipedia contributors, "Structural similarity — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Structural_similarity&oldid=1017571671, 2021. [Online; accessed 21-May-2021].

[13] Rajiv Soundararajan, "Image quality assessment." http://www.ee.iisc.ac.in/people/faculty/soma.biswas/AIP_pdf/ImageQualityAssessment.pdf.

[14] Wikipedia contributors, "Structural similarity — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Structural_similarity&oldid=1006870119, 2021. [Online; accessed 28-February-2021].

[15] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14410–14430, 2018.

[16] E. Mikhailov and R. Trusov, "How adversarial attacks work." https://blog.ycombinator.com/how-adversarial-attacks-work/. Accessed: 2017-11-02.

[17] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *arXiv preprint arXiv:1804.00792*, 2018.

[18] SECML, "Secml documentation." https://secml.gitlab.io/tutorials/03-Evasion.html, 2020.

[19] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

A.System Specifications and Experiment Environment

We use the following computational environment:

CPU: Intel Core i7-6700HQ,

GPU: NVIDIA GTX960M, 2GB, GDDR5,

RAM: 16GB DDR3,

DISK: Kingston SSD, SA400S37240G,

OS: Windows 10 Pro,

Environment: Python 3.7.5, Python 3.8; CUDA 10.2,

IDE: Anaconda 2020 - Jupyter Notebook, PyCharm 2020,

APIs: Microsoft Azure Cognitive Service Face APIs, PersonGroup[7]

Core Packages: TensorFlow 1.15.0; Keras 2.3.1; Fawkes 0.3.2.

B.Glossary

Adversarial attack is using a specific way to make machine learning algorithms make wrong decisions. The target of attacks can be models or data.

There are many types of attacks.The attacks could happen at the training time by injecting poisonous data into dataset so that the decision boundaries will be modified; the attacks could also happen at testing time, doing modifications on the testing inputs to make classifier make wrong decisions.

Azure Face Services The Microsoft Azure Face Service provides users with face-related services in the form of APIs, such as face detection, recognition, and verification.

Person Group contains different sets of face images corresponding to the sets' labels (different people). PersonGroup

Adversarial Attack An adversarial attack might entail presenting a machine-learning model with inaccurate or misrepresentative data as it is training, or introducing maliciously designed data to deceive an already trained model into making errors[11].

SSIM is the abbreviation of Structural Similarity. Structural Similarity is used in image vision to compare the similarity of two images. Structural Similarity has three main components luminance, contrast and structure[12]. Structural Similarity is different from traditional similarity comparison method like MSE (Mean Square Error) using absolute errors. Though the traditional methods are very easy to calculated, they are very poor correlation with human perception[13].

DSSIM is the abbreviation of Structural Dissimilarity. Structural Dissimilarity[14] is based on Structural Similarity [3] The formula of Structural Dissimilarity is shown below.

$$DSSIM(x, y) = \frac{(1 - SSIM(x, y))}{2} \tag{1}$$

Iterative and One-shot The attacks can be iterative or "one-shot". The iterative denotes that the computation of perturbations needed to add to the input images need multiple times to calculate. One-shot denotes that the computation will only happen once [15].

Max Iterations In Fawkes, the Max Iterations denotes the maximum times of conducting computations of gradient descent.

Learning Rate In Fawkes, Learning Rate denotes the speed of convergence of gradient descent.

Targeted Attack In Targeted Attack, the output of classifier will be a specific class. The opposite of it is Non-targeted Attack, in which the output of classifier will be a specific class [15] [16].

Clean Label Attack Clean Label Attack happens at training time. The modified inputs will be inserted into dataset to pollute dataset. The modification of inputs only happens on

inputs themselves and it will not change the label or ground truth of inputs. Clean Label Attack works well when attackers or trackers using Scrapy to collect online information (text or images) as training data [17].

Evasion Attack Evasion Attack happens at testing time. In Evasion attack, the inputs are modified to avoid detection by classifier or mislead classifier. Evasion attacks (a.k.a. adversarial examples) consists of carefully perturbing the input samples at test time to have them misclassified [18].

Adversarial Perturbation Adversarial Perturbation is the noise achieved by a specific algorithm to fool the classification or recognition system. Adversarial Perturbation should be quasi-imperceptible [15].

Adversarial Example Adversarial Example is the image or any kind of inputs that is being modified by certain perturbations in order to fool machine learning system [15].

Fooling/Success/Protection Rate Fooling/Success/Protection Rate denotes the percentage of modified images being misclassified as another class (person) or fail to recognize [15] [19] [2].

C.Experiment A

Origin Fawkes Mode	Min	Low	Mid	High
DSSIM	0.002	0.003	0.005	0.008
Origin Fawkes Value (Caucasian)	0.290	0.143	0.026	0
Origin Fawkes Value (Asian)	0.509	0.399	0.082	0.054
Origin Fawkes Value (African)	0.608	0.570	0.462	0.214
Values from Curves (Caucasian)	0.347	0.254	0.006	0.016
Values from Curves (Asian)	0.470	0.356	0.127	0.031
Values from Curves (African)	0.637	0.570	0.438	0.213

TABLE 10: Experiment A Data

D.Experiment B

DSSIM	Max Iterations	Learning Rate	Average Confidence
0.005	20	20	0.4209
0.005	20	35	0.1749
0.005	20	40	0.1413
0.005	50	20	0.0542
0.005	50	35	0.0
0.005	50	40	0.0
0.005	200	20	0.0251
0.005	200	35	0.0
0.005	200	40	0.0254

TABLE 11: Asian Female

DSSIM	Max Iterations	Learning Rate	Average Confidence
0.005	20	20	0.5696
0.005	20	35	0.4220
0.005	20	40	0.4301
0.005	50	20	0.3455
0.005	50	35	0.2305
0.005	50	40	0.1711
0.005	200	20	0.1439
0.005	200	35	0.1478
0.005	200	40	0.1646

TABLE 12: African Female

DSSIM	Max Iterations	Learning Rate	Average Confidence
0.005	20	20	0.4209
0.005	20	35	0.1749
0.005	20	40	0.1413
0.005	50	20	0.0542
0.005	50	35	0.0
0.005	50	40	0.0
0.005	200	20	0.0251
0.005	200	35	0.0
0.005	200	40	0.0254

TABLE 13: Caucasian Female

E.Experiment C

Record	DSSIM	Gender	Skin Tone	Average Confidence
1	0.002	Male	Caucasian	0.4958
2	0.003	Male	Caucasian	0.4512
3	0.004	Male	Caucasian	0.3669
4	0.005	Male	Caucasian	0.3278
5	0.006	Male	Caucasian	0.291
6	0.007	Male	Caucasian	0.252
7	0.008	Male	Caucasian	0.2471
8	0.002	Female	Caucasian	0.441
9	0.003	Female	Caucasian	0.2826
10	0.004	Female	Caucasian	0.1679
11	0.005	Female	Caucasian	0.1202
12	0.006	Female	Caucasian	0.0604
13	0.007	Female	Caucasian	0.0637
14	0.008	Female	Caucasian	0.0209

TABLE 14: C.1 Caucasian

Record	DSSIM	Gender	Skin Tones	Average Confidence
1	0.002	Male	Asian	0.5192
2	0.003	Male	Asian	0.4345
3	0.004	Male	Asian	0.3678
4	0.005	Male	Asian	0.309
5	0.006	Male	Asian	0.2022
6	0.007	Male	Asian	0.126
7	0.008	Male	Asian	0.0904
8	0.002	Female	Asian	0.526
9	0.003	Female	Asian	0.325
10	0.004	Female	Asian	0.2577
11	0.005	Female	Asian	0.1381
12	0.006	Female	Asian	0.1043
13	0.007	Female	Asian	0.084
14	0.008	Female	Asian	0.032

TABLE 15: C.1 Asian

Record	DSSIM	Gender	Skin Tone	Average Confidence
1	0.002	Male	African	0.6357
2	0.003	Male	African	0.6058
3	0.004	Male	African	0.5463
4	0.005	Male	African	0.4931
5	0.006	Male	African	0.412
6	0.007	Male	African	0.3591
7	0.008	Male	African	0.3215
8	0.002	Female	African	0.5482
9	0.003	Female	African	0.5012
10	0.004	Female	African	0.4373
11	0.005	Female	African	0.3986
12	0.006	Female	African	0.2978
13	0.007	Female	African	0.2416
14	0.008	Female	African	0.1883

TABLE 16: C.1 African

Experiment C.1 Data from Gender and Skin Tone

DSSIM \ Gender	Female	Male
0.002	0.4627	0.5502
0.003	0.3696	0.4972
0.004	0.2876	0.427
0.005	0.219	0.3767
0.006	0.1542	0.3017
0.007	0.1298	0.2457
0.008	0.0804	0.2197

TABLE 17: From Genders

DSSIM \ Skin Tone	Caucasian	Asian	African
0.002	0.4684	0.5226	0.5284
0.003	0.3669	0.3798	0.5535
0.004	0.2674	0.3127	0.4918
0.005	0.224	0.2236	0.4458
0.006	0.1757	0.1533	0.3549
0.007	0.1578	0.105	0.3004
0.008	0.134	0.0612	0.2549

TABLE 18: From Skin Tones

Method 1

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	39	40	10	0.599017
1	39	40	20	0.479899
2	39	40	30	0.436109
3	39	40	40	0.429007

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
4	39	80	10	0.507677
5	39	80	20	0.455911
6	39	80	30	0.438742
7	39	80	40	0.425394

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
8	39	120	10	0.446948
9	39	120	20	0.453926
10	39	120	30	0.463591
11	39	120	40	0.456883

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	39	40	40	0.429007
1	39	80	40	0.425394
2	39	120	10	0.446948

FIGURE 33: African Female DSSIM=0.0039

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
12	54	40	10	0.591094
13	54	40	20	0.465474
14	54	40	30	0.400913
15	54	40	40	0.332695

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
16	54	80	10	0.463465
17	54	80	20	0.401904
18	54	80	30	0.330580
19	54	80	40	0.329413

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
20	54	120	10	0.253692
21	54	120	20	0.376640
22	54	120	30	0.319202
23	54	120	40	0.315794

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	54	40	40	0.332695
1	54	80	40	0.329413
2	54	120	10	0.253692

FIGURE 34: African Female DSSIM=0.0054

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
24	65	40	10	0.596699
25	65	40	20	0.447210
26	65	40	30	0.353102
27	65	40	40	0.269017

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
28	65	80	10	0.454438
29	65	80	20	0.306124
30	65	80	30	0.229964
31	65	80	40	0.301640

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
32	65	120	10	0.375654
33	65	120	20	0.358903
34	65	120	30	0.320029
35	65	120	40	0.249744

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	65	40	40	0.269017
1	65	80	30	0.229964
2	65	120	40	0.249744

FIGURE 35: African Female DSSIM=0.0065

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
36	80	40	10	0.605680
37	80	40	20	0.476604
38	80	40	30	0.326840
39	80	40	40	0.255060

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
40	80	80	10	0.462844
41	80	80	20	0.263957
42	80	80	30	0.226457
43	80	80	40	0.181471

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
44	80	120	10	0.291774
45	80	120	20	0.247061
46	80	120	30	0.201029
47	80	120	40	0.271704

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	80	40	40	0.255060
1	80	80	40	0.181471
2	80	120	30	0.201029

FIGURE 36: African Female DSSIM=0.0080

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	45	40	10	0.659778
1	45	40	20	0.557574
2	45	40	30	0.490344
3	45	40	40	0.494456

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
4	45	80	10	0.530755
5	45	80	20	0.507382
6	45	80	30	0.495715
7	45	80	40	0.531820

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
8	45	120	10	0.510034
9	45	120	20	0.488135
10	45	120	30	0.466294
11	45	120	40	0.532009

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	45	40	30	0.490344
1	45	80	30	0.495715
2	45	120	30	0.466294

FIGURE 37: African Male DSSIM=0.0045

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
12	55	40	10	0.672575
13	55	40	20	0.577982
14	55	40	30	0.433713
15	55	40	40	0.410935

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
16	55	80	10	0.531764
17	55	80	20	0.468788
18	55	80	30	0.419016
19	55	80	40	0.424116

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
20	55	120	10	0.461212
21	55	120	20	0.485596
22	55	120	30	0.397014
23	55	120	40	0.397460

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	55	40	40	0.410935
1	55	80	30	0.419016
2	55	120	30	0.397014

FIGURE 38: African Male DSSIM=0.0055

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
24	65	40	10	0.691450
25	65	40	20	0.535849
26	65	40	30	0.487557
27	65	40	40	0.412929

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
28	65	80	10	0.575137
29	65	80	20	0.352369
30	65	80	30	0.402833
31	65	80	40	0.337592

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
32	65	120	10	0.450879
33	65	120	20	0.358671
34	65	120	30	0.413813
35	65	120	40	0.362526

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	65	40	40	0.412929
1	65	80	40	0.337592
2	65	120	20	0.358671

FIGURE 39: African Male DSSIM=0.0065

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
36	80	40	10	0.654279
37	80	40	20	0.570799
38	80	40	30	0.322128
39	80	40	40	0.377081

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
40	80	80	10	0.558624
41	80	80	20	0.354872
42	80	80	30	0.325948
43	80	80	40	0.242135

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
44	80	120	10	0.404022
45	80	120	20	0.330823
46	80	120	30	0.355733
47	80	120	40	0.257017

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	80	40	30	0.322128
1	80	80	40	0.242135
2	80	120	40	0.257017

FIGURE 40: African Male DSSIM=0.0080

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	24	40	10	0.576808
1	24	40	20	0.463990
2	24	40	30	0.420869
3	24	40	40	0.401481

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
4	24	80	10	0.397664
5	24	80	20	0.411825
6	24	80	30	0.452886
7	24	80	40	0.425798

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
8	24	120	10	0.442944
9	24	120	20	0.428813
10	24	120	30	0.431361
11	24	120	40	0.465518

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	24	40	40	0.401481
1	24	80	10	0.397664
2	24	120	20	0.428813

FIGURE 41: Asian Female DSSIM=0.0024

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
12	33	40	10	0.556122
13	33	40	20	0.336148
14	33	40	30	0.351138
15	33	40	40	0.291635

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
16	33	80	10	0.353939
17	33	80	20	0.358277
18	33	80	30	0.264251
19	33	80	40	0.329265

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
20	33	120	10	0.244176
21	33	120	20	0.225712
22	33	120	30	0.226145
23	33	120	40	0.314564

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	33	40	40	0.291635
1	33	80	30	0.264251
2	33	120	20	0.225712

FIGURE 42: Asian Female DSSIM=0.0033

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
24	47	40	10	0.582875
25	47	40	20	0.256647
26	47	40	30	0.126433
27	47	40	40	0.151024

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
28	47	80	10	0.256222
29	47	80	20	0.140165
30	47	80	30	0.207952
31	47	80	40	0.117814

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
32	47	120	10	0.151680
33	47	120	20	0.162212
34	47	120	30	0.183591
35	47	120	40	0.131781

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	47	40	30	0.126433
1	47	80	40	0.117814
2	47	120	40	0.131781

FIGURE 43: Asian Female DSSIM=0.0047

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
36	80	40	10	0.508402
37	80	40	20	0.310261
38	80	40	30	0.136411
39	80	40	40	0.067869

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
40	80	80	10	0.256694
41	80	80	20	0.087932
42	80	80	30	0.029467
43	80	80	40	0.023526

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
44	80	120	10	0.110904
45	80	120	20	0.030225
46	80	120	30	0.034871
47	80	120	40	0.012119

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	80	40	40	0.067869
1	80	80	40	0.023526
2	80	120	40	0.012119

FIGURE 44: Asian Female DSSIM=0.0080

Method 2

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	39	40	10	0.599017
1	39	40	20	0.479899
2	39	40	30	0.436109
3	39	40	40	0.429007
4	39	80	10	0.507677
5	39	80	20	0.455911
6	39	80	30	0.438742
7	39	80	40	0.425394
8	39	120	10	0.446948
9	39	120	20	0.453926
10	39	120	30	0.463591
11	39	120	40	0.456883

FIGURE 45: African Female DSSIM=0.0039

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
12	54	40	10	0.591094
13	54	40	20	0.465474
14	54	40	30	0.400913
15	54	40	40	0.332695
16	54	80	10	0.463465
17	54	80	20	0.401904
18	54	80	30	0.330580
19	54	80	40	0.329413
20	54	120	10	0.253692
21	54	120	20	0.376640
22	54	120	30	0.319202
23	54	120	40	0.315794

FIGURE 46: African Female DSSIM=0.0054

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
24	65	40	10	0.596699
25	65	40	20	0.447210
26	65	40	30	0.353102
27	65	40	40	0.269017
28	65	80	10	0.454438
29	65	80	20	0.306124
30	65	80	30	0.229964
31	65	80	40	0.301640
32	65	120	10	0.375654
33	65	120	20	0.358903
34	65	120	30	0.320029
35	65	120	40	0.249744

FIGURE 47: African Female DSSIM=0.0065

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
36	80	40	10	0.605680
37	80	40	20	0.476604
38	80	40	30	0.326840
39	80	40	40	0.255060
40	80	80	10	0.462844
41	80	80	20	0.263957
42	80	80	30	0.226457
43	80	80	40	0.181471
44	80	120	10	0.291774
45	80	120	20	0.247061
46	80	120	30	0.201029
47	80	120	40	0.271704

FIGURE 48: African Female DSSIM=0.0080

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	45	40	10	0.659778
1	45	40	20	0.557574
2	45	40	30	0.490344
3	45	40	40	0.494456
4	45	80	10	0.530755
5	45	80	20	0.507382
6	45	80	30	0.495715
7	45	80	40	0.531820
8	45	120	10	0.510034
9	45	120	20	0.488135
10	45	120	30	0.466294
11	45	120	40	0.532009

FIGURE 49: African Male DSSIM=0.0045

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
12	55	40	10	0.672575
13	55	40	20	0.577982
14	55	40	30	0.433713
15	55	40	40	0.410935
16	55	80	10	0.531764
17	55	80	20	0.468788
18	55	80	30	0.419016
19	55	80	40	0.424116
20	55	120	10	0.461212
21	55	120	20	0.485596
22	55	120	30	0.397014
23	55	120	40	0.397460

FIGURE 50: African Male DSSIM=0.0055

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
24	65	40	10	0.691450
25	65	40	20	0.535849
26	65	40	30	0.487557
27	65	40	40	0.412929
28	65	80	10	0.575137
29	65	80	20	0.352369
30	65	80	30	0.402833
31	65	80	40	0.337592
32	65	120	10	0.450879
33	65	120	20	0.358671
34	65	120	30	0.413813
35	65	120	40	0.362526

FIGURE 51: African Male DSSIM=0.0065

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
36	80	40	10	0.654279
37	80	40	20	0.570799
38	80	40	30	0.322128
39	80	40	40	0.377081
40	80	80	10	0.558624
41	80	80	20	0.354872
42	80	80	30	0.325948
43	80	80	40	0.242135
44	80	120	10	0.404022
45	80	120	20	0.330823
46	80	120	30	0.355733
47	80	120	40	0.257017

FIGURE 52: African Male DSSIM=0.0080

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
0	24	40	10	0.576808
1	24	40	20	0.463990
2	24	40	30	0.420869
3	24	40	40	0.401481
4	24	80	10	0.397664
5	24	80	20	0.411825
6	24	80	30	0.452886
7	24	80	40	0.425798
8	24	120	10	0.442944
9	24	120	20	0.428813
10	24	120	30	0.431361
11	24	120	40	0.465518

FIGURE 53: Asian Female DSSIM=0.0024

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
12	33	40	10	0.556122
13	33	40	20	0.336148
14	33	40	30	0.351138
15	33	40	40	0.291635
16	33	80	10	0.353939
17	33	80	20	0.358277
18	33	80	30	0.264251
19	33	80	40	0.329265
20	33	120	10	0.244176
21	33	120	20	0.225712
22	33	120	30	0.226145
23	33	120	40	0.314564

FIGURE 54: Asian Female DSSIM=0.0033

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
24	47	40	10	0.582875
25	47	40	20	0.256647
26	47	40	30	0.126433
27	47	40	40	0.151024
28	47	80	10	0.256222
29	47	80	20	0.140165
30	47	80	30	0.207952
31	47	80	40	0.117814
32	47	120	10	0.151680
33	47	120	20	0.162212
34	47	120	30	0.183591
35	47	120	40	0.131781

FIGURE 55: Asian Female DSSIM=0.0047

	DSSIM	MAX_ITERATIONS	LEARNING_RATE	CONFIDENCE
36	80	40	10	0.508402
37	80	40	20	0.310261
38	80	40	30	0.136411
39	80	40	40	0.067869
40	80	80	10	0.256694
41	80	80	20	0.087932
42	80	80	30	0.029467
43	80	80	40	0.023526
44	80	120	10	0.110904
45	80	120	20	0.030225
46	80	120	30	0.034871
47	80	120	40	0.012119

FIGURE 56: Asian Female DSSIM=0.0080

F.Age Range Definition

This definition is from Statistics Canada[1]

1	Children (00-14 years)
11	00-04 years
110	00-04 years
12	05-09 years
120	05-09 years
13	10-14 years
130	10-14 years

2	Youth (15-24 years)
21	15-19 years
211	15-17 years
212	18-19 years
22	20-24 years
221	20-21 years
222	22-24 years

TABLE 19: Age Definition 1

3	Adults (25-64 years)
31	25-29 years
310	25-29 years
32	30-34 years
320	30-34 years
33	35-39 years
330	35-39 years
34	40-44 years
340	40-44 years
35	45-49 years
350	45-49 years
36	50-54 years
360	50-54 years
37	55-59 years
370	55-59 years
38	60-64 years
380	60-64 years

4	Seniors (65 years and over)
41	65-69 years
410	65-69 years
42	70-74 years
420	70-74 years
43	75-79 years
430	75-79 years
44	80-84 years
440	80-84 years
45	85-89 years
450	85-89 years
46	90 years and over
460	90 years and over

TABLE 20: Age Definition 2

[1] Age Categories, Life Cycle Groupings

DSSIM	Groups	Hypothesis Test	p-Value (alpha=0.005)
0.001	Asian Female Vs African Female	$H_0 : \mu_1 = \mu_2, H_1 : \mu 1 \neq \mu_2$	0.0432
	Asian Female Vs Caucasian Female	$H_0 : \mu_1 = \mu_3, H_1 : \mu 1 \neq \mu_3$	0.00561
	African Female Vs Caucasian Female	$H_0 : \mu_3 = \mu_2, H_1 : \mu 3 \neq \mu_2$	0.000348
0.005	Asian Female Vs African Female	$H_0 : \mu_1 = \mu_2, H_1 : \mu 1 \neq \mu_2$	0.0436
	Asian Female Vs Caucasian Female	$H_0 : \mu_1 = \mu_3, H_1 : \mu 1 \neq \mu_3$	0.270
	African Female Vs Caucasian Female	$H_0 : \mu_3 = \mu_2, H_1 : \mu 3 \neq \mu_2$	1.59e-05
0.009	Asian Female Vs African Female	$H_0 : \mu_1 = \mu_2, H_1 : \mu 1 \neq \mu_2$	0.0193
	Asian Female Vs Caucasian Female	$H_0 : \mu_1 = \mu_3, H_1 : \mu 1 \neq \mu_3$	N/A
	African Female Vs Caucasian Female	$H_0 : \mu_3 = \mu_2, H_1 : \mu 3 \neq \mu_2$	0.0193

TABLE 21: Hypothesis Table