Privacy-Preserving Annotation of Face Images through Attribute-Preserving Face Synthesis

Written By:

SOLA SHIRAI

Advisor:

PROFESSOR JACOB WHITEHILL

A Major Qualifying Project
WORCESTER POLYTECHNIC INSTITUTE

Submitted to the Faculty of the Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science.

AUGUST 23TH, 2018 – MARCH 1ST, 2019

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see `https://www.wpi.edu/project-based-learning`.

Abstract

In this project we investigate the viability of collecting annotations for face images while preserving privacy by using synthesized images as surrogates. We compare two approaches: a deep learning model [1] to render a detailed 3D reconstruction of the face from an input image; and a novel generative adversarial network architecture that extends BEGAN-CS [2] to generated images conditioned on desired facial features. Using these two models, we conduct an experiment with crowdsourced workers to compare annotation quality of original face images and synthesized versions. Across 60 workers annotating a total of 180 images (60 of each version), we find that while original versions have the best accuracy (84.5%), the 3D (75.9%) and GAN (75.6%) versions show promising results.

ACKNOWLEDGEMENTS

I would like to extend my gratitude to my advisor, Professor Jacob Whitehill, for his guidance and support throughout the course of this project.

Results in this paper were obtained in part using a high-performance computing system acquired through NSF MRI grant DMS-1337943 to WPI.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

In recent years, advances in machine learning techniques have yielded great advances for image processing tasks such as image recognition [3], facial expression recognition [4], and image generation [5]. A major contributor to these advances is the use of so called *deep learning* techniques that make use of neural networks with a large number of layers. Deep learning has especially impacted tasks on images, such as classification tasks or automatic analysis of face images.

1.1 Motivation

Automatic detection and processing of human face images can be useful for a wide range of practical applications across fields such as health care [6] and education [7]. As a concrete running example of this type of application that we will use to motivate the work in this paper, consider the development of an intelligent tutoring system that changes its behavior based on student engagement like in [8]. This system will use video of a student's face, as well as various other features like the time spent solving a problem and the number of problems answered correctly, to choose what action to take that will most benefit the student. If a student's facial expression appears confused and has spent a long time solving on a problem, the system may show hints to guide them to a solution. If on the other hand a student looks bored and has correctly solved all of their problems, the system may start assigning more difficult problems. To train this model, we would collect video of a student's face, various details about their problem progression, and what actions an expert educator performs in response to a student's state. For this example, we will assume that facial expression is the most important information contained in the face images.

One could conceivably develop such a system as a neural network that utilizes face images with two supervised learning approaches (shown in Figure 1.1). In one approach (System A), we train a model that uses the face images as a feature - alongside other features like time spent solving a problem, problem accuracy, etc. - to predict the expert educator's action. This approach would simply use the face images as some high-dimensional feature vector, meaning that it would not need any sort of label associated with the facial expression for training. Given enough training data we can expect such a system to implicitly learn to detect facial expressions (e.g. confused expressions become associated with showing explanations), but for the given domain it may be difficult to collect adequate data for this implicit learning to occur.

The second approach (System B) would train a model to predict the expert educator's action in two steps. First, the face images would be processed by a neural network to determine what expression is being conveyed by the student. This facial expression, explicitly predicted by the first neural network, would then be used as a feature for predicting the expert's action. This allows System B to leverage more high-level semantic features of the face rather than only using low-level features (e.g. training on the predicted facial expression rather than on pixel values of the image). This system would require the additional data of expression labels associated with each image in order to train the first network to detect facial expressions.

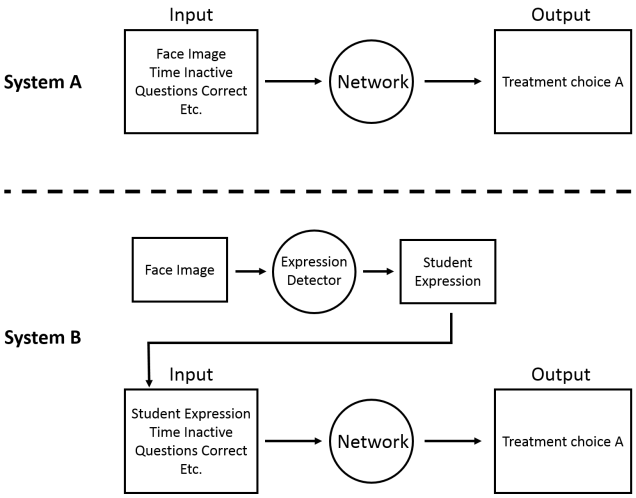

Figure 1.1: *Illustration of approaches to create intelligent tutoring system. System A passes face images as input while System B extracts the expression feature to pass as input.*

With the amount of data we can expect to collect for our example scenario, it is reasonable to suppose that the approach of System B to explicitly detect facial expressions would produce a better model. System A would require much more data to learn to map the high-dimensional feature vector (which includes the face image) to the desired action. Additionally, the explicit detection of facial expressions in System B leads to the model being much more interpretable, which could be beneficial for developers and educational researchers. Using System B would only

incur the additional cost of collecting face expression labels for the images.

To help offset the labor cost of labeling images, crowdsourcing can be a useful resource. Utilizing a crowdsourcing platform such as Amazon's Mechanical Turk (MTurk) is a common way to collect label information for images at a relatively low cost from a more diverse set of labelers [9]. However, exposing video or images of students' faces publicly could raise privacy concerns. Such privacy concerns are especially important if we consider the development of applications in other domains such as health care (e.g. images show a medical subject in pain and the subject does not wish for such images of him/herself to be made public).

It is desirable to crowdsource labels for face images so that systems can explicitly detect features of interest (like in our System B), but to do so we must perform some form of de-identification [10] to the images while retaining enough information for human workers to be able to accurately assign labels. Naive face de-identification methods such as blurring or distorting the image tend to remove too much information for human labelers to work off of.

1.2 Proposed Solutions

To collect annotations for face images while preserving privacy, we propose the approach of generating new images that share the same facial information to use as surrogate versions for workers to annotate. To preserve privacy, our aim is to reduce the amount of identity information contained in the generated images below some threshold (to make it an anonymous face) and maximize the information about facial expression that is retained.

In this project we investigate the use of two methods of generating these surrogate images. The first is to utilize 3D face models. Modern deep-learning based approaches to 3D face modeling, such as the Extreme 3D Reconstruction Project (Extreme3D) [1], are capable of taking a single image as input to produce a 3D model of the face. Extreme3D uses deep learning models to detect how to modify the expressions and shapes of the base 3D face model, as well as what pose to position the face in. Additionally, Extreme3D is able to reproduce some finer details in the face (e.g. wrinkles). This allows us to generate an image that captures the overall shape and a moderate amount of details from the original face, providing us with a surrogate version of the image that balances between retaining facial information and preserving privacy.

The second method of image generation we investigate is the use of generative adversarial networks (GANs). GANs are a type of machine learning model that works by using a pair of neural networks - a *generator* and a *discriminator* - trained in an adversarial process [11]. A GAN essentially works by having the generator try to produce fake data which closely resembles the training data while the discriminator tries to determine whether a given input datum is real or fake. Applied to face images, this leads to the creation of a generator producing fake images of faces. By using conditional training [12], we further enhance our GAN to produce face images reflective of specific input features (e.g. expression, gender, pose). Using this conditional

training we attempt to use GANs to produce fake images of faces that capture the same low-level facial features as real images, which can then be used as another type of surrogate version of the original image on which annotations can be collected.

We conduct several experiments to assess how well workers on MTurk are able to annotate images of faces using the raw images, rendered 3D face models, and generated images from our GAN model. In particular, we examine whether workers show a statistically significant difference between labeling images in their original form versus the 3D face model or generated versions of the same images.

CHAPTER 2

BACKGROUND AND RELATED WORK

In the following sections, we will discuss some key topics to provide background relevant to this project. First, key challenges that arise in de-identification of faces will be discussed. We will then introduce the topic of generating 3D face models and some key works in the field, including the Extreme 3D Reconstruction Project (Extreme3D) which we use for synthesizing 3D faces for our experiments. Finally, we will provide brief overview of GANs as well as a more focused look at a particular architecture - the Boundary Equilibrium GAN with a constrained space (BEGAN-CS) - which this project builds upon.

Before delving into these background topics, we will begin by briefly introducing *crowdsourcing*. Crowdsourcing refers to the general practice of outsourcing any kind of work to the public, or "crowd." In recent years, collecting data through crowdsourcing platforms on the internet has become increasingly accessible and commonplace, allowing researchers to focus their efforts on more important tasks. The use of Amazon's Mechanical Turk (MTurk) has been particularly widespread, providing providing researchers with a diverse population of workers to collect data from [9]. Being able to utilize the benefits of crowdsourcing in contexts where image privacy must be maintained is an important motivation of this project.

2.1 De-Identification of Face Images

De-identification refers to the general concept of removing identity information from images. For the context of this project, we want to apply some de-identification to face images before passing them on to crowdsourcing platforms. At the most basic level, this can be achieved using naive methods such as applying significant blurs, pixelization, or black boxes on faces. However, applying such naive methods will lead to a significant loss of facial information, making it

inpractical for uses such as in our project. The tasks we desire to carry out are at odds - trying to remove enough information to de-identify while retaining enough information to annotate - and as such we require the use of more sophisticated methods.

While our project takes the approaches of generating new images using 3D faces and GANs, there exist a variety of other methods for generating de-identified versions of faces. For example, the approaches using the k-same algorithm [13] combine k different images of faces that are very similar together. This leads to a resulting image that is somewhat de-identified while arguably still showing some of the important facial features from the original image. Another example of more recent work can be seen in [14], whose work aims to preserve expression of faces by detecting and preserving key facial regions (eyes, nose, mouth) and blurring the rest of the image.

2.2 Generating 3D Face Models

A key tool that has become widely utilized in modern 3D face synthesis is the use of 3D morphable models (3DMM) [15]. Techniques utilizing 3DMMs work by transforming and fitting some base 3D face model to a desired target shape. In this project, we look in particular at papers that make use of convolutional neural networks (CNNs) to fit 3D face models.

The first notable example of using CNNs for fitting 3D face models is 3D Dense Face Alignment (3DDFA) [16]. 3DDFA fits and aligns a 3DMM to the input image using cascading CNNs. Another notable example is Expression Net (ExpNet) [17]. ExpNet differed from similar works in that it performed regression directly on 3DMM expression coefficients rather than detecting and using facial landmarks. Models that generate 3D faces with greater levels of detail have also been developed. Our project looks specifically at the Extreme 3D Face Reconstruction (Extreme3D) model [1], for which we provide further details in the following section.

2.3 Extreme 3D Face Reconstruction Model Overview

For our crowdsourcing experiments involving the use of 3D face models, we chose to use the Extreme3D project [1] which is publicly available online [1]. As noted in [1], "3D face reconstruction involves the conflicting requirements of a strong regularization for a global shape vs. a weak regularization for capturing higher level details." The authors choose to approach these requirements by utilizing a combination of strong regularization for the overall face shape with weak regularization for more local details of the face. The Extreme3D model separately computes a foundation shape, facial expression, and viewpoint of a face. It then estimates a bump map - a technique to simulate bumps/wrinkles on the surface of a 3D shape - to capture mid-level features. Finally, occluded details of the face are added on to produce the final output 3D face.

[1] https://github.com/anhttran/extreme_3d_faces

Since occlusion was not a noticeable factor in the data set we used, we will not describe that aspect of their work in detail here.

First, to compute the foundation shape, s, the face is modeled by equation 2.1. Here, $\hat{s}$ is the average 3D face shape, $\alpha \in \mathbb{R}^S$ are the face shape coefficients estimated from the image, and $W^S \in \mathbb{R}^{3n+s}$ are the $S = 99$ principal components representing the distribution of 3D shapes. Extreme3D makes use of the Basel Face Model (BFM) [18] to provide $\hat{s}$ and W^S, as well as a deep learning model from [19] to regress values of α from the input image.

$$s = \hat{s} + \sum_{i=1}^{S} \alpha_i W_i^S \tag{2.1}$$

Extreme3D models face expression using the following equation:

$$e = \sum_{j=1}^{m} \eta_j W_j^E \tag{2.2}$$

In this equation, $\eta \in \mathbb{R}^m$ are expression coefficients. These expression coefficients lie in the space $W^E \in \mathbb{R}^{3n \times m}$, where $3n$ represents the 3D coordinates of vertices from BFM and $m = 29$. Equations 2.1 and 2.2 thus modifies the base face shape by computing coefficients to principal components representing the face shapes and expressions. The shapes and expressions obtained from these two equations are summed into $F = s + e$ to represent the expression adjusted foundation shape.

Finally, viewpoint is represented by $v = [r^T, t^T]$ where $r \in \mathbb{R}^3$ is the 3D rotation of the face and $t \in \mathbb{R}^3$ is a translation vector. These are computed using FacePoseNet [20], and the foundation shape is aligned to match the computed viewpoint.

Next, to add more details to the face, Extreme3D estimates local deformations of the face (rather than directly estimating the detailed face shape). These local deformations are modeled in a depth map, measuring distances of pixels in the image to the 3D face surface. First, a bump map $\Delta(p)$ is defined as follows:

$$\Delta(p) = \begin{cases} \theta(z'(p) - z(p)) & \text{face projects to p} \\ \theta(0) & \text{all other pixels} \end{cases} \tag{2.3}$$

Here, $p = [x, y]$ is a pixel in the image, $z'(p)$ is the distance from the surface (depth) of the face shape to p, $z(p)$ is the depth of the foundation shape at p, and $\theta(\dot{})$ is an encoding function. Using this information, the bump at each pixel p is computed as $\delta = z'(p) - z(p)$. The detailed depth is then computed by the following equation. The resulting detailed depth corresponds to a 3D face where each pixel p in the depth map defines a point on the detailed face shape.

$$z'(p) = z(p) + \theta^{-1}(\Delta(p)) \tag{2.4}$$

2.4 Generative Adversarial Networks

Generative adversarial networks (GANs), in their most basic form as introduced in [11], consist of a pair of neural networks that are "competing" against each other. The *generator* network takes a vector of random noise as input to generate some output data (for the sake of our context, we'll consider data as images of faces). On the other hand, the *discriminator* network takes images as input and tries to determine whether the input image is real or fake (generated). This discriminator is fed real images as well as fake images generated by the generator network and trained to determine when inputs are real or fake. The generator then is trained on the results from the discriminator, training itself based on how well the discriminator can identify generated images as fake. In this way, the two networks are pitted against each other in an adversarial fashion, with the discriminator learning to better distinguish between real and fake face images and the generator learning to generate more realistic face images to "fool" the discriminator.

GANs can also be augmented to produce class conditional images. As described in [12], simple conditional GANs can be constructed by concatenating class labels to the inputs of both the discriminator and generator. In the discriminator the addition of the class labels influences how it identifies images as real or fake, and in the generator the class label modifies the input noise to help reflect the class of images it should generate. To give an intuitive example, we can consider developing a conditional GAN for images for MNIST digits. Even if the generator creates extremely realistic images, if it doesn't produce images of the correct class the discriminator will be able to easily identify that the images are fake. This encourages the generator to not only produce realistic images but also to create images that align with the input conditions.

In practice, training GANs to produce high quality high quality images is difficult. One of the most prominent issues that can be seen in developing GANs is known as *mode collapse*. When this occurs, the images produced by the generator collapse onto some mode seen in the training data (e.g. only producing images of one type of number when training on the MNIST dataset). Various improvements to GAN architectures and loss functions have been developed to inhibit such behaviors [11, 21], but fully avoiding mode collapse often relies on empirical trial-and-error through tuning the network's architecture and hyperparameters.

Another example of failure is when the discriminator trains too quickly. A variety of factors may influence the discriminator to become very good at determining whether images are real or fake. If the discriminator is *too* good at its task, it will always correctly determine when generated images are fake. This subsequently causes the gradient for the generator to go to 0, making it unable to continue learning to generate better images.

2.4.1 Boundary Equilibrium GAN

Of the wide assortment of GAN architectures that have been developed, in this project we choose to make use of the boundary equilibrium GAN (BEGAN) architecture from [22]. Unlike traditional GANs, the discriminator network of BEGAN is an auto-encoder. Rather than having the discriminator trained on a binary classification task (identifying if images are real or fake), in BEGAN the discriminator is trained on the reconstruction error of input images (with the target of minimizing reconstruction error for real images and maximizing it for fake images). The use of auto-encoders and reconstruction loss were first proposed in the energy-based GAN (EBGAN) model [23], where reconstruction loss allowed for a diverse set of gradients for the networks to train more efficiently than with a binary loss.

BEGAN expands upon the EBGAN model by introducing an equilibrium enforcing term to balance out the weighting of reconstructing real and generated images. As explained in [22], in the early stages of training the generator has low reconstruction error because the generated data is close to 0 and the real data distribution hasn't been learned by the discriminator. The equilibrium term in BEGAN ensured stable training by maintaining that the reconstruction loss of real images always has a greater influence than that of generated images.

Yet another improvement upon the BEGAN architecture is BEGAN with a constrained space (BEGAN-CS) [2]. This model introduces a constraint on the internal state of the auto-encoder, limiting the difference between it and the input noise for generated images. The addition of this constraint is shown to improve mode collapse in the BEGAN-CS model. In this project, we further build upon BEGAN-CS's architecture to introduce conditional training.

2.5 BEGAN-CS Architecture Overview

The final GAN architecture used for this project primarily builds upon the architecture of BEGAN-CS [2]. This architecture involves the use of a generator $G(z)$, which takes a noise vector z as input to produce and output image, and a "discriminator" auto-encoder network $D(x)$, which takes an image x to reconstruct by passing through its encoder ($Enc(x)$) and decoder ($Dec(x)$).

The following equations describe the objective function of BEGAN-CS. Here, $\mathscr{L}(v)$ represents the reconstruction loss of the auto-encoder.

$$\mathscr{L}(v) = |v - D(v)|^2 \quad \text{where} \quad \begin{array}{ll} D : \mathbb{R}^{N_x} \mapsto \mathbb{R}^{N_x} & \text{is the auto-encoder function} \\ v \in \mathbb{R}^{N_x} & \text{is a sample of dimension } N_x \end{array} \tag{2.5}$$

Next, the $\mathscr{L}_D$ term is the loss function for the discriminator network and the $\mathscr{L}_G$ term is the loss of the generator, both of which we are trying to minimize.

$$\mathscr{L}_D = \mathscr{L}(x_{real}; \theta_D) - k_t \cdot \mathscr{L}(G(z_G; \theta_G); \theta_D) + \alpha \cdot \mathscr{L}_C, \quad \text{for } \theta_D \tag{2.6}$$

$$\mathscr{L}_G = \mathscr{L}(G(z_G;\theta_G);\theta_D), \quad \text{for } \theta_G \tag{2.7}$$

$$\begin{aligned} &\mathscr{L}_C = \|z_D - Enc(G(z_D))\|, && \text{the constraint loss} \\ &k_{t+1} = k_t + \lambda(\gamma\mathscr{L}(x;\theta_D) - \mathscr{L}(G(z_G;\theta_G);\theta_D)), && \text{for each epoch} \end{aligned} \tag{2.8}$$

Following from the architecture of the BEGAN [22], the loss function of BEGAN-CS includes the term k_t which helps to stabilize the training process by maintaining a balance between the reconstruction loss of real and generated data such that $\mathscr{L}(x) > \mathscr{L}(G(z))$. Newly introduced by BEGAN-CS is the latent-space constraint loss term, $\mathscr{L}_C$, which enforces that the internal state of the encoder for generated data, $Enc(G(z))$, resembles the original input noise into the generator. γ and α are hyperparameters.

The use of this auto-encoder architecture empirically has shown to give us higher quality generated images. Additionally, the addition of the constraint loss $\mathscr{L}_C$ drastically improves the model's ability to avoid severe mode collapse. To achieve the purpose of our project, the next step is to expand this architecture to condition the generator and discriminator on some form of class labels.

CHAPTER 3

METHODOLOGY

The primary motivation of this project involves the use of some type of image modification or generation to serve as a surrogate image to use to collect annotations on crowdsourcing platforms. As such, the main goal of our experimentation will be to produce surrogate images and compare the quality of annotations on new images with those of the original face images. Given that synthetic images have effectively erased identity information, an ideal finding from our experimentation would be that workers show no significant difference in the annotations they give to original images versus the surrogate images, meaning that the use of those surrogate images is a viable method of collecting annotations of face without revealing the original images to the public.

This section will provide a general overview of the methodologies used in this project. We will cover the general experimental design and concepts surrounding how we will analyze results. Additionally, we provide some insight into image selection and generation using 3D face models and GANs. Finally, we provide a brief summary of the methodology used in exploring and developing the GAN model used to generate our images.

3.1 Experimental Design

3.1.1 Data

For the various stages of this project, we used the following datasets of face images: LFW [24], AFLW [25], and GENKI [26–28]. The LFW and AFLW datasets contained a large number of face images (roughly 40,000 total) and were primarily used in exploration of the 3D face model generation and development of the GAN models. Images from the GENKI datasets were used for training the final GAN model, and the GENKI-4K dataset in particular was used for selecting

images to perform crowdsourcing experiments on. We chose to use the GENKI-4K dataset for experiments because they provided ground-truth labels of whether the face was smiling or not-smiling (whereas the other datasets didn't contain annotations related to facial expression or emotion).

Separately, images from MNIST [29] were also used in the development of the project's GAN model. MNIST images are fairly small, which allowed for faster training times. Additionally, empirical evaluation of the results generated when trained with conditioning was easier with images of numbers rather than faces.

3.1.2 Experiment Setup

To conduct our annotation experiment, we used the crowdsourcing platform of MTurk due to its popularity and existence of various studies supporting the validity of its data collection. Labeling all images took roughly 10 minutes, and workers were given a compensation of $1 for completing the task.

Based on the available data and considerations of the difficulty of the task, we chose to conduct our annotation experiment on only one facial feature: whether or not the displayed face was smiling. This allowed for us to have a ground-truth label (by using sample images from the GENKI-4K dataset) to compare the accuracy of annotations. Because the task is quite simple, we expect the vast majority of workers on MTurk to have enough natural perceptual expertise to recognize smiles in facial images. In our experiment, we compared the performance of annotations given on the original, 3D face model, and GAN-generated versions of images.

While the task for our experiment was quite simple, it is still important to consider that the various images would inherently have differences in the difficulty of correctly annotating them. To address this, we chose to 1) show every worker the exact same set of images (original, 3D face model, and GAN-generated versions) and 2) generate 3D face model and GAN images for each original face image in the experiment set. The first point addresses differences in worker ability, ensuring that we do not end up in a situation where highly skilled workers happen to annotate easy images and low-skilled workers annotate difficult images. Our second point addresses the differences in image difficulty; since workers are essentially annotating three different versions of the "same" image, if our proposed solution is valid we can expect that image difficulty will be similar for each version of the image.

To select our images and collect labels for use in our GAN model, we used a facial analysis software Emotient [30]. Emotient provides a wide variety of analytics for a face image, including the probability that the face is female or male, the amount of "joy evidence" displayed in the image, and the yaw, pitch, and roll of the face. In an effort to try to have a sampling of images from the GENKI-4K dataset that contained a variety of hard- and easy-to-label images, we selected the images to use in our experiment based on joy evidence. We binned all GENKI-4K

images based on the joy evidence detected by Emotient into bins of size 0.5 (where the majority of evidence scores lied between -3 and 3). From each of these bins, we sampled an equal number of images from each bin for our experiment image set. The distributions of smiles in our set of images selected using this method was 33.3%.

3.1.3 Evaluation

After the workers completed the task on MTurk, we had response data containing the label - smile or nonsmile - given by each worker for each image. To compare the performance of each of the three versions of images with this data, we performed the following evaluations:

One Sample T-Test on Difference of Accuracy: For the original and 3D face versions of each image, we counted the number of workers who provided the correct annotation. For each image, we then subtracted the counts of correct annotations given to the 3D face version from the counts of the original version. This gives us a list of differences in worker annotation accuracy for each image. On this list of differences, we ran a one sample t-test with a null hypothesis that the mean is 0 - i.e. if we can reject the null hypothesis, worker annotation accuracy is significantly different on original and 3D face versions of the same set of images. This process was repeated to compare the difference between original and GAN-generated versions as well as between GAN-generated and 3D face versions of the images.

Majority Vote Accuracy: As majority vote is often used for actually determining what label to give to data based on crowdsourcing, it is important to consider this in our evaluation as well. Using the collected data, for each version of each image we selected the annotation given by the majority of workers as the majority vote annotation. We then took the overall accuracy over the entire set of the annotations given by majority vote for the three versions to compare their accuracy. We also performed some analysis into the accuracy of majority vote for images where 75% and 90% of the workers gave the same annotation. While these methods lack the statistical power of the previous evaluation, they do provide important insight into how these methods may be used empirically.

Comparing Accuracy to Joy Evidence: Based on the joy evidence detected by Emotient, we can suppose that images with evidence values near 0 may be considered more difficult to distinguish between smiling or not. To look into whether this assumption holds true, we visualized our results to compare the joy evidence and worker annotation accuracy for each image.

3.1.4 Power Analysis Simulations

To assess the usefulness of collecting annotations on generated images, we need to analyze whether there are statistically significant differences in annotation correctness between original and generated images. Because running a crowdsourcing task on MTurk requires us to pay some compensation to workers, it was important to perform simulations and power analyses to optimize

the cost-benefit of performing our experiment. Our power analysis was used to determine the number of unique workers to have complete our task and the number of images to have each worker annotate.

In order to choose the number of workers N and images M to use in our experiments, we chose to conduct simulations and measure how often we could detect statistical significance. This can be done by generating two sets of numbers of length M, representing whether or not the worker correctly labeled the image, for each of the N workers. These two sets would be summed up across the N workers and subtracted, resulting in a final set of M numbers representing the difference of accuracy between the two versions of images. A one sample t-test was then run on this list of differences with a null hypothesis of 0 to determine whether statistical significance was observed. For each combination of N and M, we repeat this simulation process multiple times and record the probability that we detect statistical significance for each combination.

An issue that now arises is that we are trying to perform simulations over a distribution that we do not know the mean or variance of, making it tricky to generate our two sets of numbers for each worker. However, even without having a good estimate of the true mean or variance, we still can perform the simulation if we *assume* that the two versions of images will have different performance. In our simulations, for each worker we generated data from a binomial distribution with a success probability of 0.6 for one set and 0.65 for the other. To relate this to the actual situation we are performing the simulation for, this would mean that workers correctly label 60% of images from one version and 65% from another. By performing the simulations with the assumption that the two versions do in fact have different labeling accuracy, we can use our simulation to determine values of N and M that will maximize the likelihood that we detect this difference.

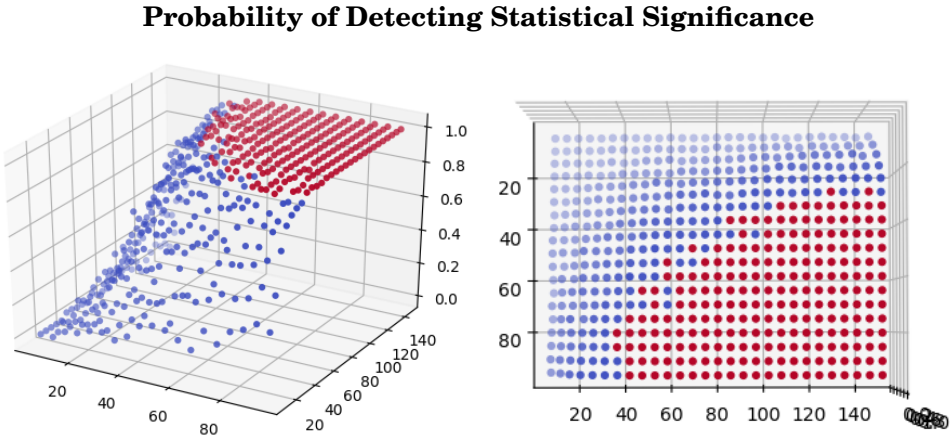

Figure 3.1: *Simulation results for the number of workers and images (X and Y axis) versus the probability that statistical significance was detected (Z axis). Left shows a view of the values plotted in 3D space, right shows a top-down view.*

Figure 3.1 shows results of our simulations. We ran simulations for N in increments of 5

between the values of 5 and 100 and for M in increments of 5 between 10 and 150. The probability of detecting statistical significance is plotted on the Z axis, and worker/image combinations that detected statistical significance with a probability of over 0.95 are colored red. based on the results from our simulations, we can see that an arc of high probabilities of observing a statistically significant difference exists around ranges where the product of worker and image count is 3,600 (e.g. N=80 and M=40, N=60 and M=60).

After completing these simulations, we now turn our attention to another consideration for our experiment – cost. There are no concrete rules that dictate the amount of pay that workers on MTurk should receive for any task. While our task of labeling smile or nonsmile requires basically no expertise by the worker, we still must consider the amount of time needed to complete the task and provide a fair compensation to ensure that the quality of our data is good. While we did attempt some calculations of cost for combinations of N and M, our analysis ultimately could not provide a concrete "best" answer because the determination of cost between increasing N and M was fairly arbitrary. We ultimately chose to conduct our experiment on 60 workers and 60 images (of which we had 3 versions, so 180 images total per worker), using a guideline of paying $6 per hour that is often followed for MTurk tasks.

3.2 3D Face Model Experiments

In this project, we primarily explored the use of two previously developed models to generate 3D faces: 3DDFA [16, 31] and Extreme3D [1]. For both of these systems, we used pretrained models provided by the authors of their respective papers to generate our images.

Our initial experimentation with 3D face models was to determine which one to ultimately use in our crowdsourcing experiment in MTurk. Both 3DDFA and Extreme3D generate what appear to be high quality 3D face reconstructions. Faces generated by Extreme3D contained more fine details in the face (e.g. wrinkles) because of its use of bump maps, while 3DDFA did not. This extra level of detail arguably could defeat the purpose of trying to generate images to preserve privacy. On the other hand, it is possible that the level of detail in 3D faces generated by 3DDFA are not adequate for workers to properly identify the facial expressions we are interested in annotating.

We conducted a small scale experiment using 7 subjects and 40 distinct images sampled form the AFLW dataset. For each of these 40 images, we generated two versions of 3D faces using 3DDFA and Extreme3D. On this overall set of 120 images, we asked the subjects to choose which of the following eight emotions were displayed by the image: joy, sadness, surprise, contempt, anger, fear, disgust, or neutral. The number of subjects that correctly labeled each image were tallied up, and we ran a one sample t-test on the differences in accuracy between the original and 3D versions of images (similarly to our evaluation approach described in section 3.1.3). The 3D

faces generated by Extreme3D showed much better performance than 3DDFA in our experiment, and based on those results we decided to use Extreme3D for our crowdsourcing experiment.

The actual process for generating images for the crowdsourcing experiment using Extreme3D was very straight forward. Using the Extreme3D model provided by the authors, we passed in our 60 selected images to have 3D face versions generated. We ensured during the selection of our 60 images that all images could be properly processed by the Extreme3D model.

3.3 GAN Experiments

To generate images for our crowdsourcing experiment using a GAN, we made use of facial information reported by Emotient [30]. As noted earlier, we used the probability that the face was male, the amount of "joy evidence" displayed in the image, and the yaw, pitch, and roll of the face in training our GAN. To train our final GAN model, we used images from the GENKI dataset [26]. The images used for training did not include images in the GENKI-4K dataset, which we selected our images from for the crowdsourcing experiment. The model was trained for 150 epochs using WPI's high performance computing system.

After training our GAN, we used the same set of 5 facial features to generate the GAN versions of images to conduct our experiment. Using our 60 selected images, we generated new images by passing in the 5 features. In principal, we can expect that the generated images would reflect these 5 features, and thus would provide viable surrogate images to perform crowdsourcing on.

3.3.1 Exploring GAN Architectures

In the process of developing our final GAN model, we performed a wide range of exploration of GAN architectures and hyperparameters. Some of the main challenges encountered in the development of an effective GAN were image quality and mode collapse. Image quality was influenced by several factors including GAN's architecture, loss function of the GAN, depth of the neural network, and number of training images. Analysis of which of these factors happen to cause the most influence for any particular model is quite difficult, especially since image quality largely depends on manual inspection and subjective judgement. Furthermore, simply trying to make more complex or deeper networks for the GAN can become highly resource intensive.

Mode collapse is another important challenge that was encountered frequently in our exploration of GANs. While methods of avoiding or mitigating mode collapse exist, there is yet no absolute solution to completely avoid it for GANs. From the perspective of conducting experiments on generated faces, mode collapse is especially problematic since it reduces the expressiveness of our model. A major benefit of using a GAN as opposed to just modifying a base face image is that we can generate a range of different types of faces; if we only could generate a few types of

images for certain conditions, this benefit would be completely lost. Mode collapse would also be easy for crowdsourced workers to notice, and may cause bias in their annotations.

Because much of the evaluation of image quality required manual inspection, we made use of the MNIST [29] dataset for early experimentation. This allowed us to make use of smaller GANs that required less training time. Additionally, producing images of numbers as opposed to faces made judgements of their quality much easier, especially when exploring conditional models (whether or not an image is the number 1 or not is easier to determine than whether a face is "smiley" or not).

The GAN architectures explored in this project include the basic, fully connected GAN [11], GANs with improved training metrics presented in [21], deep convolutional GANs [5], basic conditional GANs [12], and the boundary equilibrium GAN (BEGAN) [22]. Face images produced by the BEGAN architecture empirically gave us the highest quality images during our exploration phase, leading us to pursue adding onto this architecture for our project. Since BEGAN does not support conditional image generation in its basic setup, we performed further experimentation and development to produce the final conditional BEGAN model for our project (detailed in Chapter 5).

CHAPTER 4

PROPOSED ARCHITECTURE: CONDITIONAL BEGAN-CS

For our second proposed solution to the problem of crowdsourcing annotations for face images while preserving privacy, we use a GAN to produce entirely new faces that share some low-level features as the original images that we want to annotate. To do this, we develop an extension to the architecture of BEGAN-CS to utilize conditional training in order to control the facial features of our generated images.

4.1 Conditional Image Generation

To enable conditional training on labels into the BEGAN-CS architecture, we chose to introduce an auxiliary predictor for labels into the discriminator network. This auxiliary network is a simple fully connected network that takes the internal state of the encoder as input and outputs predictions of labels. Additionally, in the generator, we concatenate the label information with the input noise.

To accommodate this new predictor, the loss functions for the generator and discriminator have a new loss added. Since the labels for our dataset were continuous values, we chose to use mean squared error (MSE) to measure the error of our predictor. Given input ground-truth labels y and predicted labels $\hat{y}$, the loss functions are then updated with an MSE loss term $\mathscr{L}_{MSE}$.

$$\begin{aligned} \mathscr{L}_D &= \mathscr{L}(x_{real};\theta_D) - k_t \cdot \mathscr{L}(G(z_G|y;\theta_G);\theta_D) + \alpha \cdot \mathscr{L}_C + \mathscr{L}_{MSE}(y;\hat{y}_{real}), && \text{for } \theta_D \\ \mathscr{L}_G &= \mathscr{L}(G(z_G|y;\theta_G);\theta_D) + \mathscr{L}_{MSE}(y;\hat{y}_G), && \text{for } \theta_G \end{aligned} \tag{4.1}$$

Figure 4.1 and 4.2 provide a visual aid to help understand the final architecture of the GANs. The generator concatenates a vector of random noise z and labels y as input to generate an image.

The discriminator takes either a generated image x_{fake} or real image x_{real} as input, encodes the image into a vector $\hat{z}$, and produces a reconstructed image $D(x)$ and a predicted label $\hat{y}$.

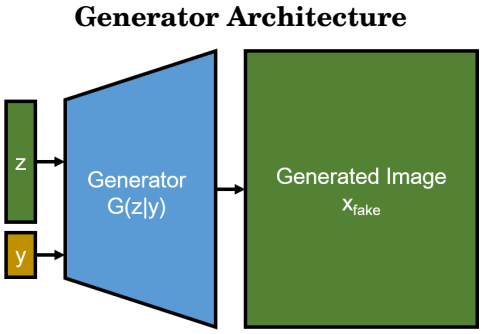

Figure 4.1: *Overview of the final generator network.*

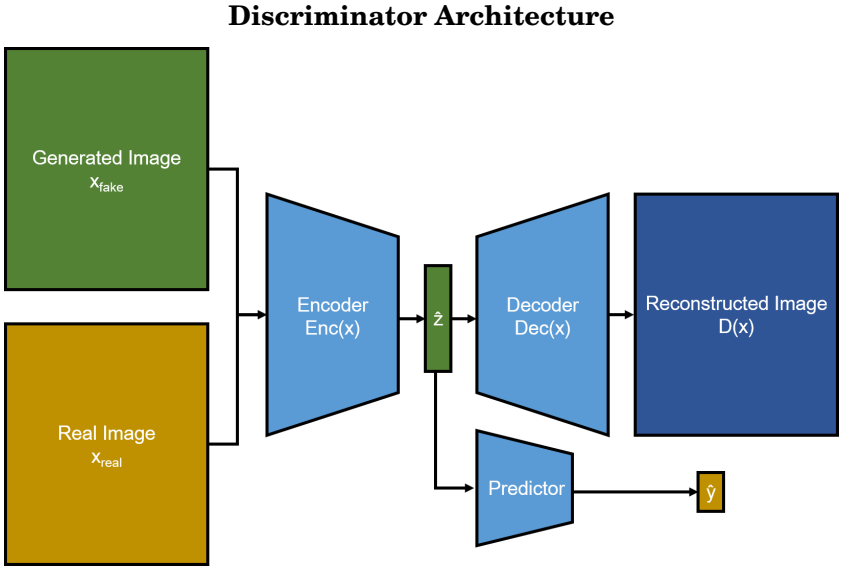

Figure 4.2: *Overview of the final discriminator network. Note that only one of the inputs x_{real} and x_{fake} are passed through the network at a time.*

In the discriminator's loss function, we only use the MSE on label predictions over real images to allow the auxiliary network to learn to more accurately predict labels from the internal state of the auto-encoder. The generator's loss function is amended to include the MSE on labels predicted by the auxiliary network on the generated images. Intuitively, our discriminator's training objectives are 1) minimize reconstruction error for real images, 2) maximize reconstruction error for fake images, 3) minimize difference between generator input z and the encoding $\hat{z}$ of the generated image, and 4) minimize the difference between the real label y and predicted label $\hat{y}$ for real images. This setup will encourage the generator to produce images that reflect the input labels in a similar fashion to traditional conditional GANs; even if the generator is

producing extremely realistic faces, the discriminator will learn to penalize fake images because the predicted labels will not match the ground truth.

4.2 Model Hyper-parameters

We trained our final GAN model on 64 x 64 images with 5 dimensional labels representing "joy evidence," probability that the face's gender is male, yaw, pitch, and roll. As a preprocessing step, these labels were normalized to values between 0 and 1. This normalization step was partially chosen empirically and partially to help balance out the influences of reconstruction and label prediction losses.

Input into the generator network was a 64 dimensional noise vector, sampled uniformly between -1 and 1. Concatenating this with input labels, the input was passed through convolutional layers and upsampled to form the 64 x 64 image outputs. The Adam optimizer was used to train the discriminator, with a learning rate of 0.0001.

In the discriminator network, we used convolutional layers in the encoder to encode the input 64 x 64 image into a 64 dimensional vector. The decoder portion of the discriminator had an identical shape to the generator network (other than inputting labels). The auxiliary predictor was a 2 layer fully connected network, taking the image encoding as input and producing label predictions. We set the values of α and γ to 0.5 and 0.1, respectively, and we once again used the Adam optimizer with a learning rate of 0.0001 for the training process.

4.3 Generated Image Examples

In this section, we share some examples of images generated by our final GAN model after 150 epochs of training on 64 x 64 images from GENKI datasets. First, figure 4.3 shows an example of a group of images generated by our network. These images were generated using a sampling of real labels, and show us a variety of different face poses, genders, and smiles. We can see that the images generally have good quality, although we can also see that some images appear fairly distorted (especially around the eyes of the generated images).

Next, we compare real images to images generated using the same feature labels. Figure 4.6 shows examples of 8 images (in the top row) followed by 3 images generated using their gender, joy, and pose feature labels. The smile and gender features appear to be reflected fairly well in our examples, although we can see that images with higher joy evidence produce generated images with very "toothy" smiles compared to the original images (as can be seen in the second column of examples). Additionally, the pose information of generated images seems to only weakly resemble the original images.

An important aspect of our GAN is the ability to specify and control certain features of the generated faces to reflect features found in original images. Figure 4.5 shows samples of

Sample of Generated Images

Figure 4.3: *Example of a collection of images generated by our final GAN network. Each generated image is of size 64 x 64 and use some real label from the training set.*

Comparing Generated to Real Images

Figure 4.4: *Examples of real images and images generated using the features detected from real images. The top row are real images, while the 3 rows beneath each image shows 3 examples of generated images using the same labels with different input noise vectors.*

generated images where we interpolate the joy evidence and male gender probability labels that we pass into the generator network. Note that all other inputs are kept consistent while interpolating these two features. These two features appear to be getting generated quite well by

our GAN, with more female-like faces appearing thinner and very differences between smiles and non-smiles being displayed.

Interpolation of Generated Images across Gender and Joy

Figure 4.5: *Examples of generated images when the gender and joy labels are modified. Left to right shows increased joy evidence and top to bottom shows increased male gender probability.*

We also show interpolation across face poses in figure 4.6. This figure shows combinations of interpolating the feature labels across yaw, pitch, and roll. Each of these three values is interpolated across the values of 0 to 1, where each value was normalized from the degrees of yaw/pitch/roll into the range of (0, 1). Unlike interpolating across gender and joy, our control over face pose seems to be somewhat lackluster. While we can see some evidence of control over the yaw of the face, the effects of controlling pose are much more subtle. These observations are also reflected in our comparison of real images to generated images in figure 4.4.

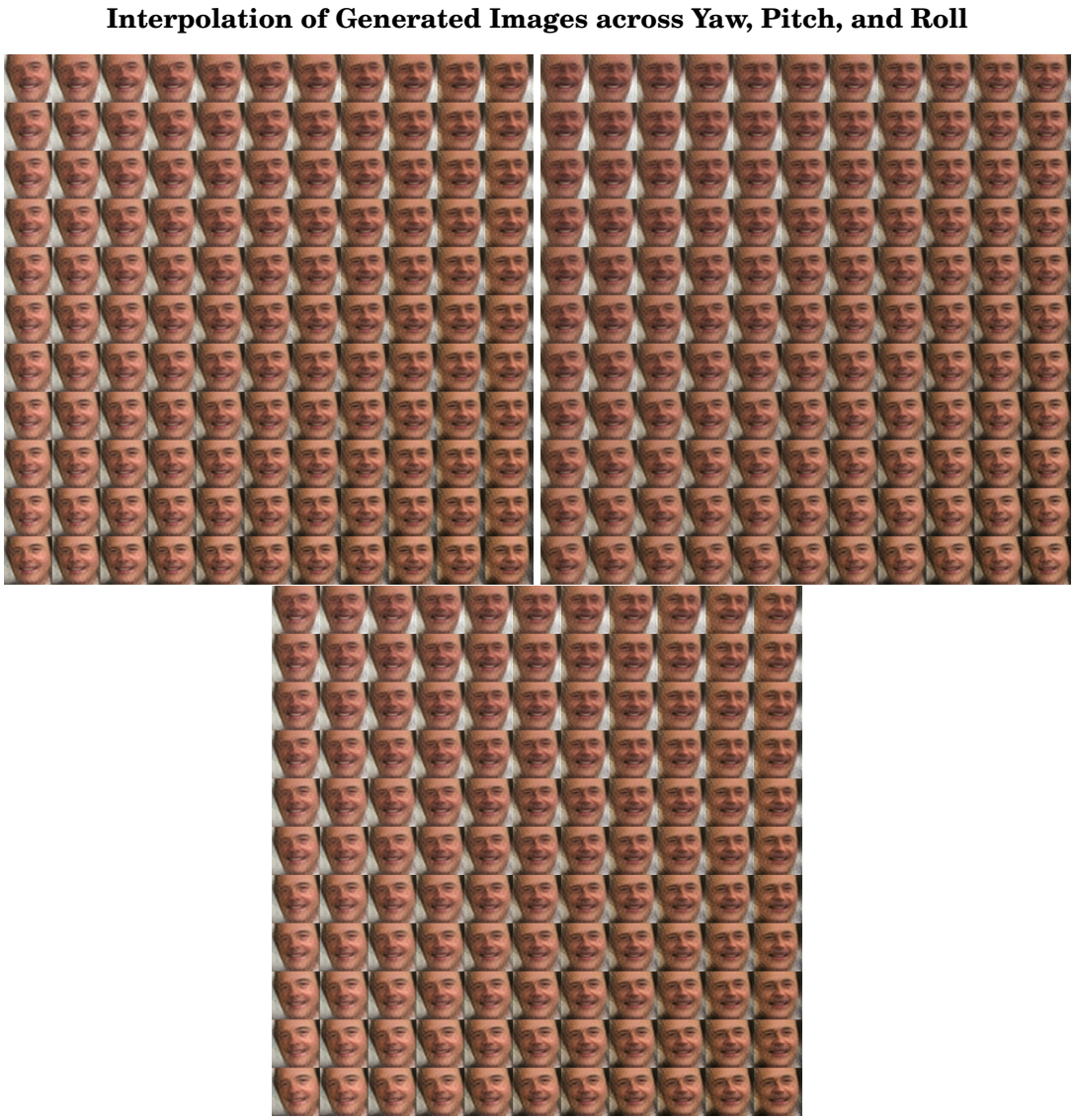

Figure 4.6: *Images of interpolating yaw and pitch (top left), pitch and roll (top right), and yaw and roll (center) of generated images.*

CHAPTER 5

EXPERIMENTAL RESULTS

The following section details results gathered from our experiment conducted on MTurk. Annotation data was collected from 60 unique workers tasked to label a total of 180 images as "smile" or "nonsmile." The 180 images consisted of 60 images sampled from the GENKI-4K dataset, 60 images generated by the Extreme3D model (E3D), and 60 images generated by our GAN based on the gender probability, joy evidence, yaw, pitch, and roll detected by Emotient for each image. Accuracy / correctness of annotations in this experiment are given with respect to the ground-truth labels provided in the GENKI-4K dataset.

5.1 Overview of Worker Performance

Table 5.1 shows an overview of accuracy results obtained from our experiment. Average accuracy gives us the average number of workers that correctly annotated each image while majority vote accuracy gives us the accuracy of annotations when we take the majority vote of workers as the label. We also observe the accuracy of each version when we only consider the majority vote for images with high worker consensus on which label to give. We can see from these results that both types of generated images show poor performance in comparison to their original versions.

We can also gain some insight into how majority vote accuracy changes for each of the three versions as more workers contribute to the vote. Figure 5.1 shows majority vote accuracy when we take a sample of N workers from the dataset (averaged over 10 trials for each N). We can see that the accuracy for each version becomes quite stable after sampling around 40 workers. We also can visually observe here how the original versions of images consistently outperform both generated versions.

Image Version	Avg. Accuracy	Majority Vote Accuracy	Maj. Vote Acc. (75% Consensus)	Maj. Vote Acc. (90% Consensus)
Baseline	83.3	83.3	-	-
Original	84.5	91.2	92.0	97.6
GAN	75.6	80.0	82.4	85.7
E3D	75.9	85.0	87.5	89.6

Table 5.1: *Accuracy of worker labels provided for each type of image. Baseline accuracy is given based on Emotient "Joy Evidence" of the image (where evidence > 0 was given a "smile" label and < 0 was given "nonsmile"). Average accuracy is given as the average number of workers who provided the correct annotation to each image in the set divided by the total number of workers. Majority vote accuracy gives the average accuracy of labels given to images in each set if when the majority vote of workers is taken for each image; the % consensus versions of the majority vote measure label accuracy only for images where X% of workers gave the same annotation.*

Majority Vote Accuracy for Samples of Workers

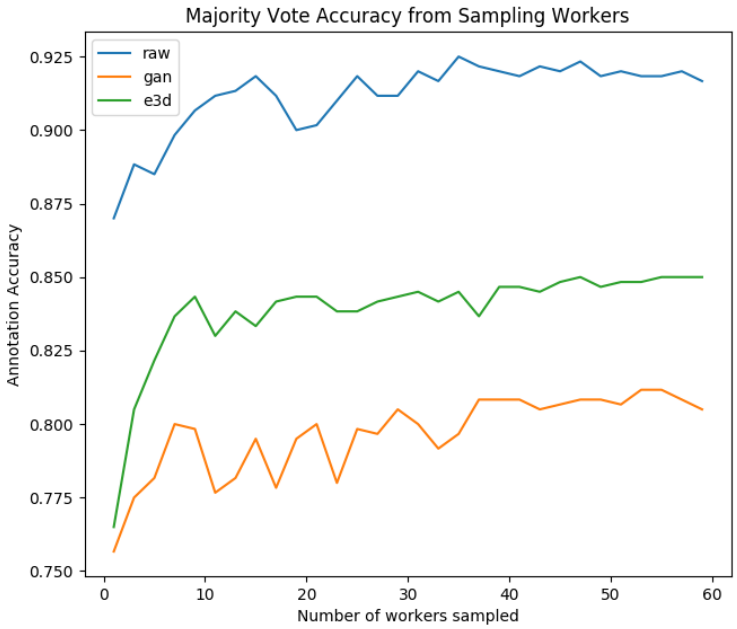

Figure 5.1: *Majority vote accuracy for each version of images when we sample a number of workers. Sampling started at 1 worker and increased in increments of 2, up to 59 workers.*

Unfortunately, the results of our experiment show that annotation accuracy does not show extremely promising results. Even compared to the baseline - which we take as using Emotient's "joy evidence" that we used to select the images - has a greater annotation accuracy than GAN and Extreme3D versions of images. When we take the majority votes, the Extreme3D versions of images do manage to perform better than the baseline, but the GAN still only reaches 80% accuracy.

Versions	t-statistic	p-value
Original - GAN	2.919	0.0049
Original - E3D	2.647	0.0104
GAN - E3D	0.076	0.9393

Table 5.2: *Results of one sample t-test on the differences of accuracy. Note that the differences taken here are the number of correct labels for each image rather than the accuracy represented as a percentage.*

5.2 T-Test on Differences of Accuracy

To better analyze the results highlighted above, we performed a one sample t-test on the differences of annotation accuracy for each of the three versions. Specifically, for a pair of image versions, we subtracted the number of workers that correctly annotated version A of an image from the correct annotations of version B of the same image. Using a list of such differences for all 60 images, we performed a one sample t-test with a null hypothesis of 0 to determine whether the observed differences were statistically significant or not. Table 5.2 shows the results of our t-test on the differences between original and GAN, original and Extreme3D, and GAN and Extreme3D.

Based on these results, we can determine that the difference in performance between the original and generated images is statistically significant for both images generated by our GAN and the Extreme3D faces, at a significance level of over 99%. With a t-statistic of around 3, this means that the accuracy of annotations on our generated versions is roughly 5% worse than on original versions.

5.3 Accuracy vs Joy Evidence

An important factor to consider in the results observed from our crowdsourcing experiment is the fact that we used Emotient to both make selections of images to use in our experiment and to train our GAN model. Since our images were selected by sampling from bins of joy evidence, it is possible that we can consider images that had evidence near 0 as more "ambiguous" images that workers had difficulty annotating correctly. Additionally, we can expect any sort of error in joy evidence detected by Emotient to become strongly reflected in the GAN, which likely will lead to large errors in annotations given to those images.

Figure 5.2 plots worker annotation accuracy for each image compared to that image's joy evidence detected by Emotient. We can see that for both the original and GAN versions of images, accuracy tends to decrease for images with evidence near 0. This pattern is especially distinct for GAN images, where images with joy evidence between -1 and 1 suffer from extremely low annotation accuracy. However, GAN images do appear to have very high accuracy for images very large or small joy evidence values. On the other hand, Extreme3D versions of images seem

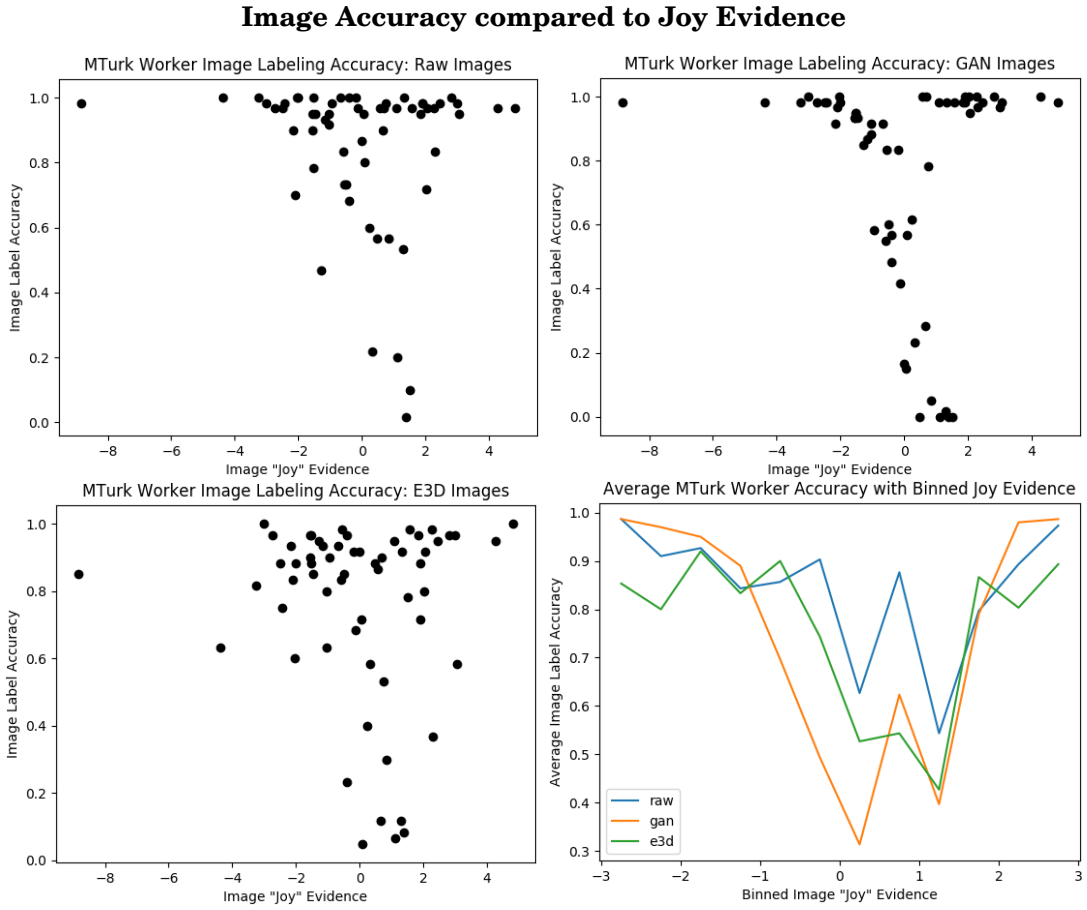

Figure 5.2: *Worker annotation accuracy for each image compared to its Emotient joy evidence. Accuracies are given for original ("raw") versions in the top-left, GAN versions in the top-right, and Extreme3D versions in the bottom-left. The graph in the bottom-right shows average worker accuracy for sets of images binned by their joy evidence.*

to have a larger scattering of accuracy. We can observe that some images with evidence high evidence (e.g. an image with joy evidence of about 3) only has an accuracy of around 60%. To help gather some more insight into this, we can look to the bottom-right graph in figure 5.2. This graph plots the average accuracy for images grouped together by their joy evidence (using bins of size 0.5). We can see that the GAN accuracy shows a steep decline near 0, but at the outer edges the accuracy is greater than the raw and 3D versions. The GAN version performs especially poorly in the (0, 0.5) range, although it does perform nearly as well as the 3D version in the range of (0.5, 1.5).

These results highlight the challenge that are faced when trying to use crowdsourced workers to label our more ambiguous images. Images that were slightly more ambiguous in what facial expression was displayed - and consequently had Emotient provide joy evidence that was closer to 0 - were, for the most part, unable to be accurately annotated by workers.

5.4 Example Images and Worker Performance

Here we will present some examples of images on which MTurk workers performed particularly poorly for one or more of the image versions. First, in figure 5.3, we can see an example of images that had poor annotation accuracy for all versions. The ground truth given by GENKI-4K for this image is that it is a nonsmile, but the accuracy of workers correctly labeling it as nonsmile were 8%, 0%, and 2% for the 3D, GAN, and original versions, respectively.

Image Example: Low Accuracy

Figure 5.3: *Extreme3D, GAN, and original version of image with poor annotation accuracy.*

Next, figure 5.4 gives an example of an image which had high worker accuracy for the 3D face version of the image but poor accuracy on the other two. The ground truth label for this image was nonsmile. Looking at the original image, on the right, it is easy to see how worker accuracy may have been poor in deciding whether the face showed a smile or not, and only 6 of the workers labeled it correctly. Because Emotient detected the original image as having a moderate amount of joy evidence, the image generated by the GAN shows a very clear smile; none of the MTurk workers labeled this image as nonsmile. On the other hand, the Extreme3D version of this image had 47 workers correctly label (78% accuracy) it as nonsmile.

Image Example: High Accuracy on Extreme3D

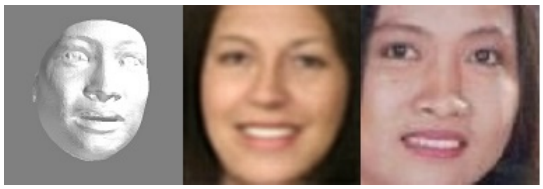

Figure 5.4: *Extreme3D, GAN, and original version of an image with high accuracy for Extreme3D version.*

Figure 5.5, on the other hand, shows a face that had very poor performance for the 3D face. The accuracy for the Extreme3D version of this image was 5% compared to the 80% for original and 57% of GAN versions. The fact that the original image had sunglasses likely affected these results heavily, as the generated 3D face reproduces some lines on (where the glasses were) that appear to make the image seem "angry" when the ground truth label is that the face is smiling.

Cases where the GAN versions of images performed very poorly were often caused by images with very slightly positive joy evidences. Figure 5.6 shows a face where 0 workers correctly

Image Example: Low Accuracy on Extreme3D

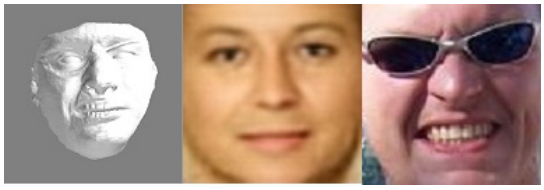

Figure 5.5: *Extreme3D, GAN, and original version of an image with high accuracy for Extreme3D version.*

labeled the GAN version of the image as a nonsmile. The joy evidence used as input for this image was fairly low, but in this situation it seems that the GAN happened to produce an image that looks particularly smiley.

Image Example: Low Accuracy on GAN

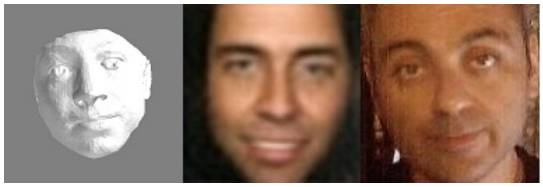

Figure 5.6: *Extreme3D, GAN, and original version of an image with high accuracy for Extreme3D version.*

Chapter 6

Conclusion

In this project we explored the viability of using synthesized face images to collect annotations while preserving privacy of the original faces. While annotation of generated versions is less accurate than original, our results show promise in using similar approaches to conduct crowdsourcing using surrogate images to preserve privacy. In our experiment, where workes were tasked to label whether or not the face was smiling, we find that crowdsourced workers on MTurk correctly annotate generated versions of images with accuracies of 75.6% and 75.9% for images generated by our GAN model and the Extreme3D model, respectively, compared to the 84.5% accuracy on the original versions of images. Aggregating worker annotations using majority vote increases accuracy to 80.0%, 85.0%, and 91.2% for GAN, Extreme3D, and original versions of images, respectively.

For the development of our GAN model, we successfully demonstrate a method to introduce conditional image generation into the BEGAN architecture. The addition of our auxiliary prediction network shows to successfully motivate the generator to produce images that align with the desired features used as input, allowing it to generate face images that reflect features from original images for use in crowdsourcing. Furthermore, the use of our GAN model to generate surrogate images allows us to completely preserve privacy.

6.1 Reflections

At a personal level, the process of completing this Major Qualifying Project was a valuable learning experience. The ability to adapt and expand the project's focus was important throughout its duration, as limitations of current systems were revealed or intermediate results failed to strongly support hypotheses. Although having a thorough, well researched plan that is diligently

followed can certainly make for a great project, it was equally valuable to gain experience in shifting and expanding the project's focus. This project also strongly reemphasized the importance of attention to detail; on more than one occasion, progress of training GANs or analyzing results was inhibited because data labels were incorrectly matched. For future students, I strongly recommend learning from this project's shortcomings by conducting more thorough research of related work in the early stages of the project and taking more care in the organization and processing of data.

6.2 Future Work

While our GAN showed successful results, it still has much room for improvement. Two points in particular that could be improved are improved tuning of the weighting of reconstruction loss versus MSE loss and the use of more facial features in training. We found that the use of MSE loss for our conditional architecture caused an increase in instances where the generated images were highly blurry or distorted, likely caused by the generator network placing too much importance on producing images that could have their input features correctly predicted rather than reconstructing high quality faces. Investigating methods to balance our the relative weight of these two losses (e.g. by starting the MSE loss as having very little weight and slowly increasing it over training) would likely be beneficial to produce better face images.

Our second point of improvement, the use of more facial features, is targeted at producing better face images in situations where joy evidence is near 0. While the original face's joy evidence may be ambiguous, it is possible that some other emotion (e.g. anger) is being displayed prominently in the image. When our network only uses joy evidence, we would not be taking such other emotions into consideration and cause us to fail to produce images that accurately reflect all of the facial information displayed by the original image. Further exploration into using a wider range of emotion information or other detected details about the original face images would likely allow us to create GANs that can create better surrogate images while still completely preserving privacy.

Another topic of future work is to expand the experiments conducted in this project to a more complex task. Due to the scope of this project, we were unable to conducted highly detailed investigations into the use of 3D faces or GANs to produce images for more complicated tasks such as annotating what emotion is shown in an image. Such experiments would help provide further evidence for the viability of this approach of collecting annotations.

BIBLIOGRAPHY

[1] Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gérard G. Medioni.
Extreme 3d face reconstruction: Looking past occlusions.
CoRR, abs/1712.05083, 2017.

[2] Chia-Che Chang, Chieh Hubert Lin, Che-Rung Lee, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen.
Escaping from collapsing modes in a constrained space.
CoRR, abs/1808.07258, 2018.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
CoRR, abs/1512.03385, 2015.

[4] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong.
Facial expression recognition via a boosted deep belief network.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[5] Alec Radford, Luke Metz, and Soumith Chintala.
Unsupervised representation learning with deep convolutional generative adversarial networks.
CoRR, abs/1511.06434, 2015.

[6] J. Thevenot, M. B. López, and A. Hadid.
A survey on computer vision for assistive medical diagnosis from faces.
IEEE Journal of Biomedical and Health Informatics, 22(5):1497–1511, Sep. 2018.

[7] Abdolhossein Sarrafzadeh, Hamid Gholamhosseini, C Fan, and Scott Overmeyer.
Facial expression analysis for estimating learner's emotional state in intelligent tutoring systems.
pages 336– 337, 08 2003.

[8] Beverly Woolf, Winslow Burleson, Ivon Arroyo, Toby Dragon, David Cooper, and Rosalind Picard.
Affect-aware tutors: Recognizing and responding to student affect.

IJLT, 4:129–164, 01 2009.

[9] Amazon.
Mechanical turk.
`http://mturk.com`.

[10] Ralph Gross, Latanya Sweeney, Jeffrey Cohn, Fernando De la Torre, and Simon Baker.
Face De-identification, pages 129–146.
07 2009.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio.
Generative adversarial nets.
In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[12] Mehdi Mirza and Simon Osindero.
Conditional generative adversarial nets.
CoRR, abs/1411.1784, 2014.

[13] Elaine M. Newton, Latanya Sweeney, and Bradley Malin.
Preserving privacy by de-identifying face images.
IEEE Trans. on Knowl. and Data Eng., 17(2):232–243, February 2005.

[14] G. Letournel, A. Bugeau, V. T. Ta, and J P. Domenger.
Face de-identification with expressions preservation.
In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4366–4370, Sep. 2015.

[15] Volker Blanz and Thomas Vetter.
A morphable model for the synthesis of 3d faces.
In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*, pages 187–194, 1999.

[16] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li.
Face alignment in full pose range: A 3d total solution.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[17] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gérard G. Medioni.
Expnet: Landmark-free, deep, 3d facial expressions.

CoRR, abs/1802.00542, 2018.

[18] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter.
A 3d face model for pose and illumination invariant face recognition.
In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, Sep. 2009.

[19] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni.
Regressing robust and discriminative 3d morphable models with a very deep neural network.
CoRR, abs/1612.04904, 2016.

[20] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gérard G. Medioni.
Faceposenet: Making a case for landmark-free face alignment.
CoRR, abs/1708.07517, 2017.

[21] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen.
Improved techniques for training gans.
CoRR, abs/1606.03498, 2016.

[22] David Berthelot, Tom Schumm, and Luke Metz.
BEGAN: boundary equilibrium generative adversarial networks.
CoRR, abs/1703.10717, 2017.

[23] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun.
Energy-based generative adversarial network.
CoRR, abs/1609.03126, 2016.

[24] Gary B. Huang and Erik Learned-Miller.
Labeled faces in the wild: Updates and new reporting procedures.
Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.

[25] Martin Koestinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof.
Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization.
In *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[26] `http://mplab.ucsd.edu`.
The MPLab GENKI Database.

[27] `http://mplab.ucsd.edu`.
The MPLab GENKI Database, GENKI-4K Subset.

[28] http://mplab.ucsd.edu.
The MPLab GENKI Database, GENKI-SZSL Subset.

[29] Yann LeCun and Corinna Cortes.
MNIST handwritten digit database.
2010.

[30] iMotions.
Emotient.
https://imotions.com/emotient/.

[31] Xiangyu Zhu Jianzhu Guo and Zhen Lei.
3ddfa.
https://github.com/cleardusk/3DDFA, 2018.