Project Final Report

MirrorMe

12/05/2020

Chang-Won Lee (1005185103), Vo Yagong (1005173688), Esther Yoo (1005187557)

Word Count: 1997
Penalty: 0%

Introduction

Facial expression is a ubiquitous method of communication that is instrumental in conveying messages and emotion. Precise control over facial expression is essential not only in real life, but also in media such as film and video games.

Through our project – MirrorMe – we attempt to extend our users' control of facial expression over virtual media.

MirrorMe attempts this by using neural networks to transfer an individual's expression in one image to an individual in another image. Our goal is to preserve precision in nuanced expressions while allowing for a breadth of targets to which the expression may be transferred. Possible target individuals may include the expression's originator, other persons and animated characters.

Several areas may benefit from facial expression transfer. One beneficiary is the animation industry, which requires the imitation of human expressions by animated characters. Expression transfer can provide animators with these imitations for use as reference, or in the final work. This will allow for more precise, emotive expressions in media such as films and video games. In addition, expression transfer can be used in social media filters and entertainment tools.

Generally, facial expression transfer is difficult. Manual implementation requires expertise in photo editing tools. Additionally, it is unfeasible to automate using traditional programming due to the breadth of possible expressions and faces. Neural networks can provide a robust approach to automation that produces accurate results.

Background & Related Work

A Compact Embedding for Facial Expression Similarity[1]:

The team produced a model that maps an image of an expression to a vector that can be used for facial expression comparison.

This demonstrates that it is possible to extract meaningful information about facial expressions that is partially independent of the originator of the expression and can be encoded in a tractable format.

Unconstrained Facial Expression Transfer using Style-based Generator[2]:

A neural network is used to extract information about a face and encode it into a StyleGAN[2] style vector containing information about facial appearance and expression.

Linear programming is used to fuse the style vectors of an expression source image and a target image. StyleGAN then constructs an image from the resulting vector. The result maintains the appearance of the target with the expression of the source.

This demonstrates that robust facial expression transfer has been accomplished for humans.

Illustration/Figure

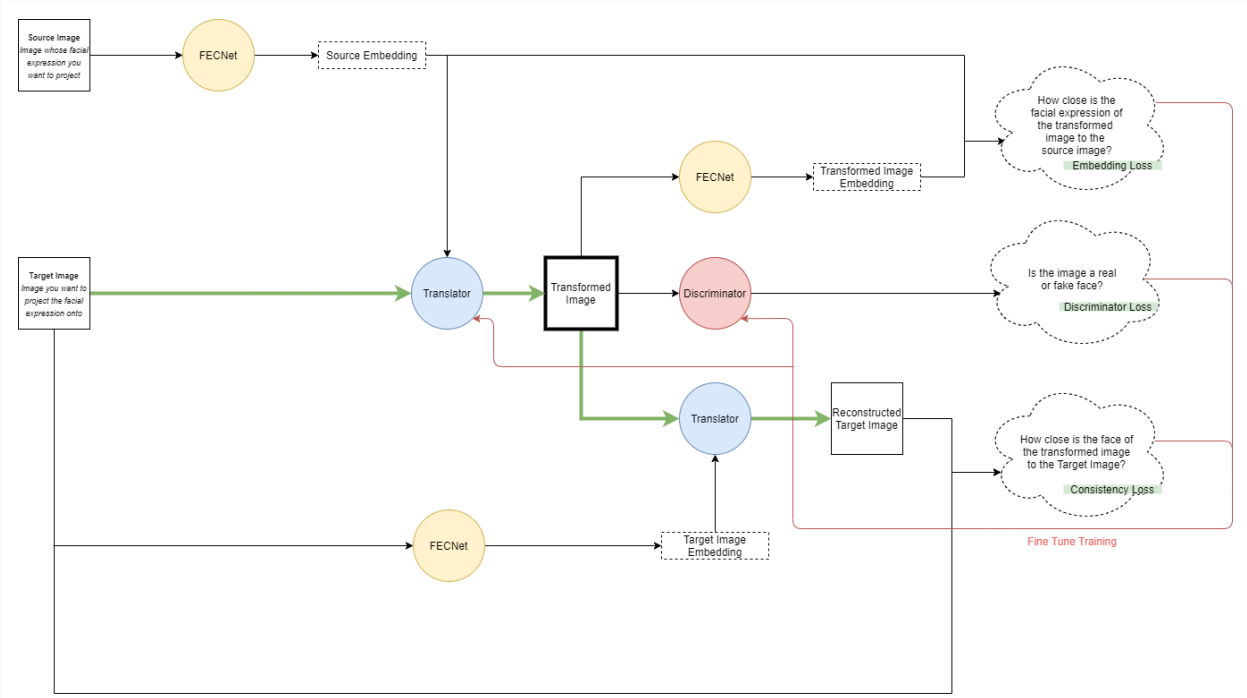


Figure 1. Overall training loop and model architecture

Data and Data Processing

Google Facial Expression Comparison[3]

Data: 500,203 image triplets of human faces.
Label: Labelled by six annotators, indicating which of three images had the least similar expression.

Figure 2. A sample triplet, with the least similar expression outlined in red

(Link1, *bounding box coords,
Link2, *bounding box coords,
Link3, *bounding box coords,
Triplet type (one-class, two-class, or three-class – we describe this later)
Annotator1 id, Annotator1 vote,
Annotator2 id, ...)

Figure 3. FEC Dataset Sample Format

Collection and Cleaning:

1. Filter to include only triplets that annotators "strongly agreed" on, ie. majority vote made up at least ⅔ (recommended by the FEC paper[1])
2. Download image links and crop to the bounding box coordinates specified.
3. Discard triplets containing non-downloadable images.

Result:
308,923 triplets corresponding to 75,135 unique images (due to image overlap between triplets).

The general similarity of expressions in these triplets is a relevant statistic. Training on triplets that have similar expressions can increase the network's ability to make comparisons on small details.

The images in the dataset were sampled from another dataset divided into classes of expressions (eg. Amusement, Anger). The authors noted three types of triplets and specified this type in the .csv file.

- *One-class triplets*: All the three images share a category label, see Figure 2(a). These triplets are useful for learning a fine-grained expression representation.
- *Two-class triplets*: Only two images share a category label and the third image belongs to a different category, see Figure 2(b). As explained in Section 3, images sharing a category label need not form the (visually) most similar pair in these triplets.
- *Three-class triplets*: None of the images share a common category label, see Figure 2(c). These triplets are useful for learning long-range visual similarity relationships between different categories.

Figure 4. The three classes of images [1]

The dataset is approximately balanced between these three types.

One-class	Two-class	Three-class	Total
107761	101640	99522	308923

iCartoonFace [4]

Data: 389,678 images of cartoon character faces and bounding box for each face.
Images were cropped according to these specifications.

Figure 5. Samples of cartoon faces from the dataset

Further Data Collection

To allow FECNet to compare expressions between humans and cartoon characters, we trained it on triplets sampled from a pool containing images from both the FEC and iCartoon datasets to a total of 464,813 images. We developed an annotation program to facilitate the sampling and labelling of these triplets.

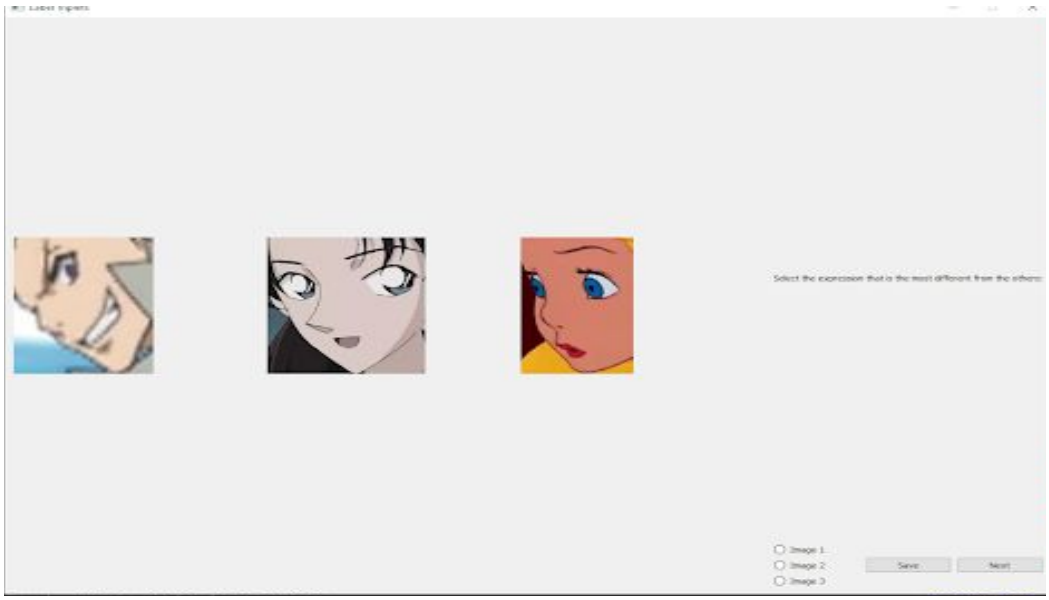


Figure 6. A screenshot of our developed labelling program

We compared on the basis of expression alone, ignoring factors such as gender, pose, and type (human/animated).

Each of us labelled 10,000 triplets. Triplets were composed mostly of cartoons, as the pool contains 389768 cartoons and 75135 humans. For balance, we reused 45,000 human triplets from the FEC dataset, skewing the balance towards human faces (it was deemed favourable to prioritize performance on humans).

We used the following sample format, taking the majority vote for the 45,000 FEC triplets as the label:

(Image1, Image2, Image3, label)

Final Dataset: 75,000 triplets
Test:Validation:Training Split: 0.1:0.2:0.7.

FECNet Triplet Data Processing
The following transformations were applied for the input FECNet images:

```
tf = transforms.Compose([
    transforms.Lambda(lambda x:Image.open(x).convert("RGB")),
    transforms.ToTensor(),
    transforms.Resize((self.size,self.size))
])
```

GAN Data Processing

The GAN needs two input images at a time. To accomplish this, a custom GAN_Dataset class was created to load both target/source, then concatenated using a custom ConcatDataset class. After applying the transformations, the dataloader function outputted source and target image batches, ready for input to the GAN.

```
transform = transforms.Compose([
                transforms.ToTensor(),
                transforms.Normalize((0.5,0.5,0.5),
                                     (0.5,0.5,0.5)),
                transforms.Resize((224,224)),
                transforms.RandomRotation(45),
                ])
```

Architecture

There are three parts to our architecture:

1) The network to create the expression embedding (“FECNet”),
2) The generator (“Translator”),
3) The discriminator

All models used normal weight initialization.

FECNet

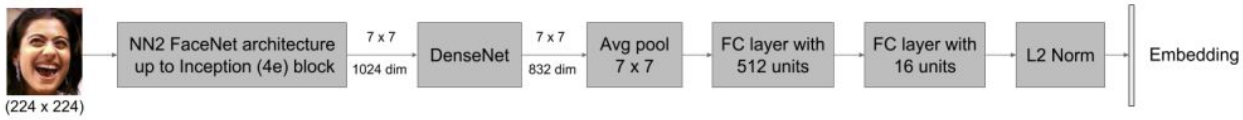


Figure 7. FECNet architecture

Our model utilized a pre-trained FECNet implemented as described by [1]. We used transfer learning to train further using our custom dataset.

We used the triplet loss function [1], where I_x refers to each image in the triplet and e_{Ix} refers to the embedding of I_x produced by FECNet. For the most similar pair (I_1, I_2), the triplet loss encourages the distance between e_{I1}, e_{I2} to be smaller than the distance to e_{I3} by comparing the distance of each e_{I1},e_{I2} to e_{I3} and e_{I1},e_{I2} against 0 by a maximum function.

$$l(I_1, I_2, I_3) = max(0, \|e_{I_1} - e_{I_2}\|_2^2 - \|e_{I_1} - e_{I_3}\|_2^2 + \delta) + max(0, \|e_{I_1} - e_{I_2}\|_2^2 - \|e_{I_2} - e_{I_3}\|_2^2 + \delta),$$

where δ is a small margin.

Figure 8. Triplet Loss function [1]

Translator: UNet

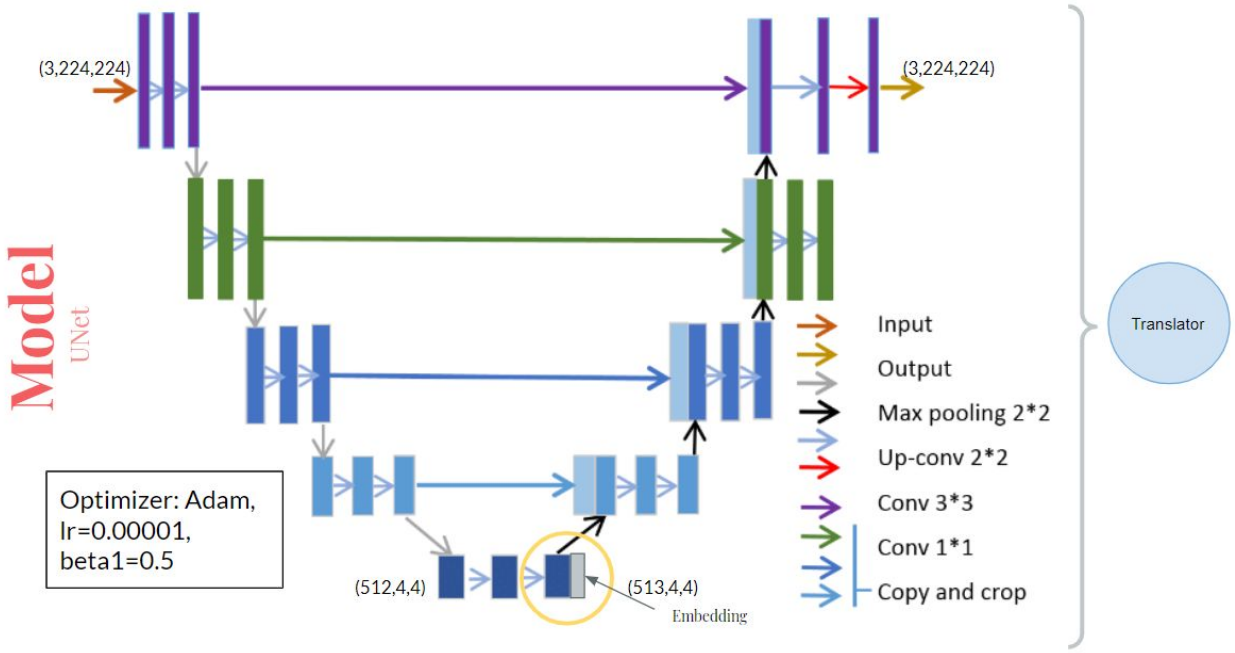


Figure 9. Schematic representation of the UNet architecture implemented in MirrorMe

Left: a series of convolutional layers to downsample the input image to a 4x4 feature map with 512 channels. We use a stride=2 to downsample (no pooling).
Bottom: concatenate the embedding to the feature map.
Right: upsample the modified feature map with transposed convolutional layers to reconstruct the input image with the facial expression represented by the embedding vector.

Our architecture contains only 1 convolutional/transposed convolutional layer per block instead of 3 layers and 6 overall levels instead of 5.

Discriminator: PatchGAN

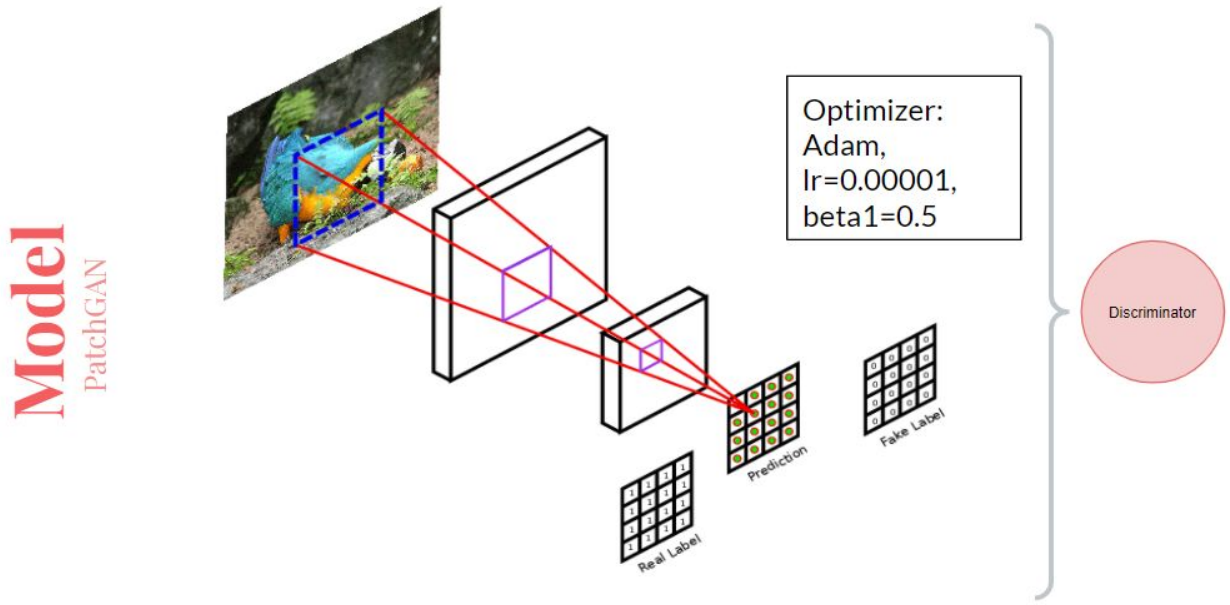


Figure 10. Schematic representation of the PatchGAN architecture (CycleGAN)

The PatchGAN discriminator determines whether or not the image produced by the translator represents a real human face. The discriminator classifies each 28x28 patch of input images as real (label=1) or fake (label=0) in a manner similar to convolution.
The predictions made by PatchGAN are used to inform the effectiveness of the translator -- its successful training will direct the translator to produce better outputs until the discriminator is consistently fooled between a real/fake image, resigning the discriminator to a random binary classifier that achieves 50% accuracy.

Loss Function

$$Loss = \alpha \times embedding\ loss + \beta \times adversarial\ loss + \gamma \times consistency\ loss$$

α, β, γ: penalty weights for each type of loss; increasing its value would increase the significance of the associated loss in the objective function.

Embedding Loss: Quantify closeness of transformed and source image embeddings; measures expression similarity (MSELoss)

Adversarial Loss: Loss of the discriminator in attempting to classify the transformed image as true/false (MSELoss)

Consistency Loss: compare the resemblance between the target image and the reconstruction of the target image from the transformed output (L1). The translator should be able to regenerate the original image. This loss is a unique characteristic of a CycleGAN.

Baseline Model

Description: Detect faces in the source and target images and swapping the faces using OpenCV was used to accomplish this task.

This performs expression transfer, but does not maintain facial appearance. It does not require a NN; a simple computer vision operation is sufficient.

Figure 11. Baseline result

Quantitative Results

FECNet

Our training of the FECNet yielded the following accuracy plots:

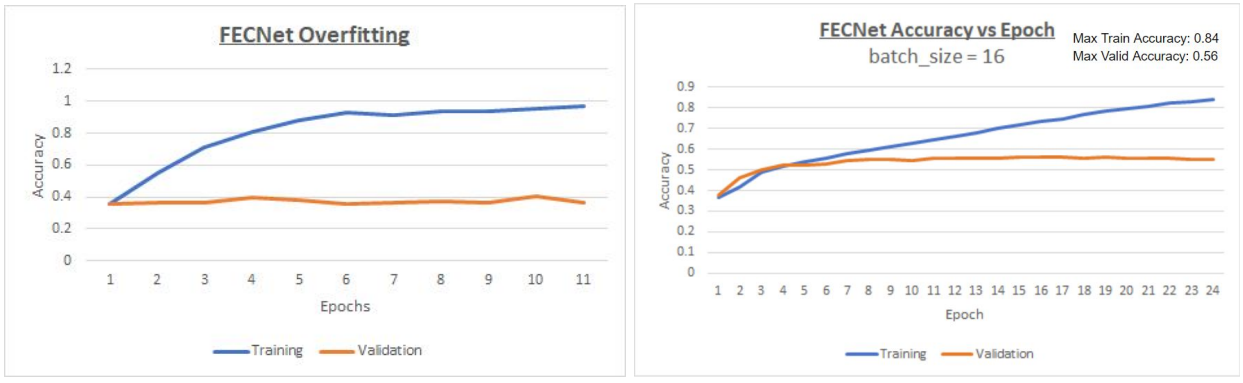


Figure 12. Accuracy plots for FECNet overfitting (right), best (left)

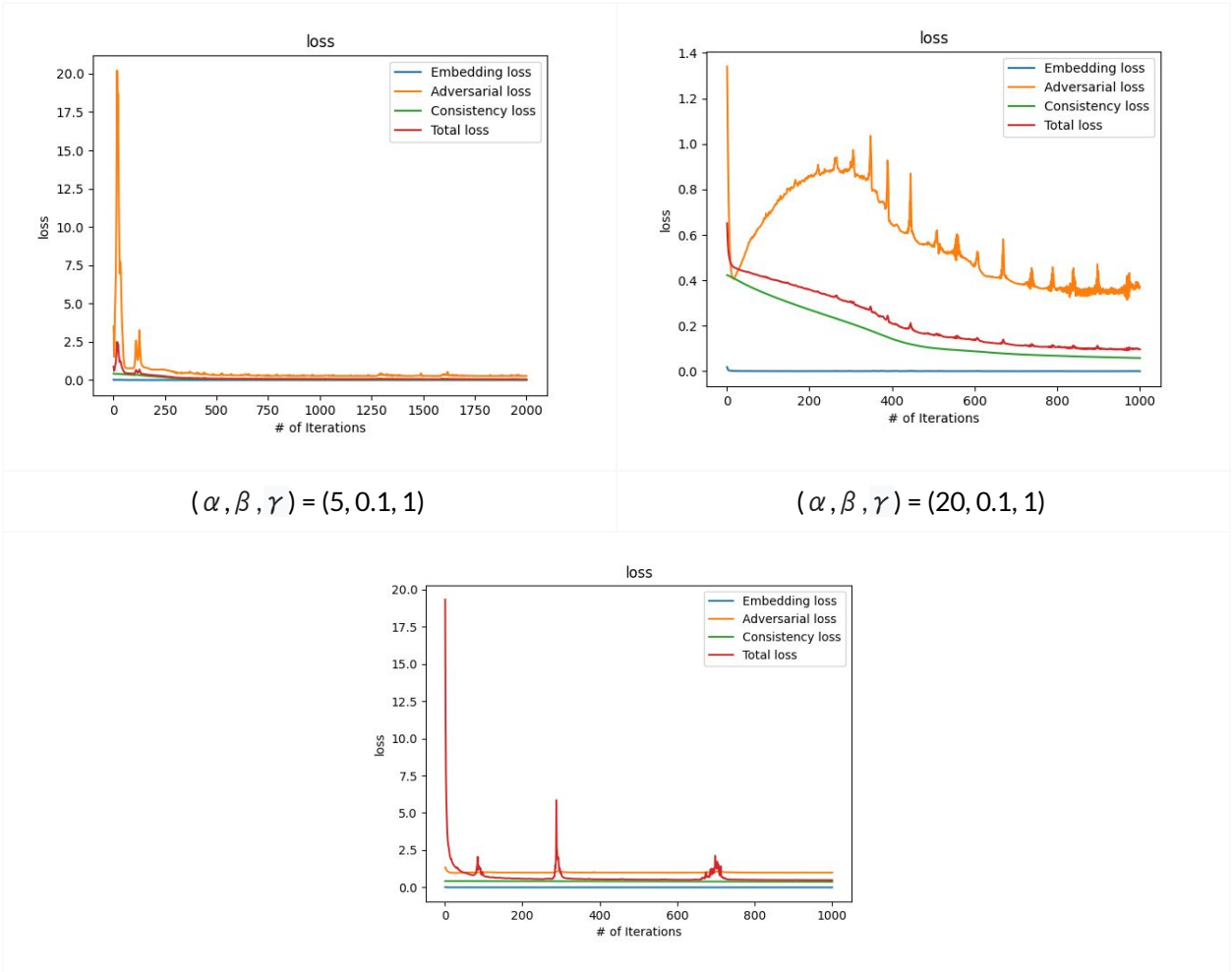
We defined the accuracy metric for the FECNet as the percentage of triplets for which the network correctly identifies the least similar expression.

The model overfit on a small sample of 4000 images to 100% accuracy. Upon training on the whole dataset, the validation accuracy plateaued at 56%, while training accuracy continued to rise. We decided to stop training early to prevent further overfitting.

Note that the accuracy of random selection is 33%, so learning did occur. However, poor performance on the validation set indicates that FECNet failed to generalize well.

Full MirrorMe

All individual losses and the total decreased throughout the training loop. Embedding loss was consistently the lowest, despite using different values for loss penalty weights.



$(\alpha, \beta, \gamma) = (5, 0.1, 1)$

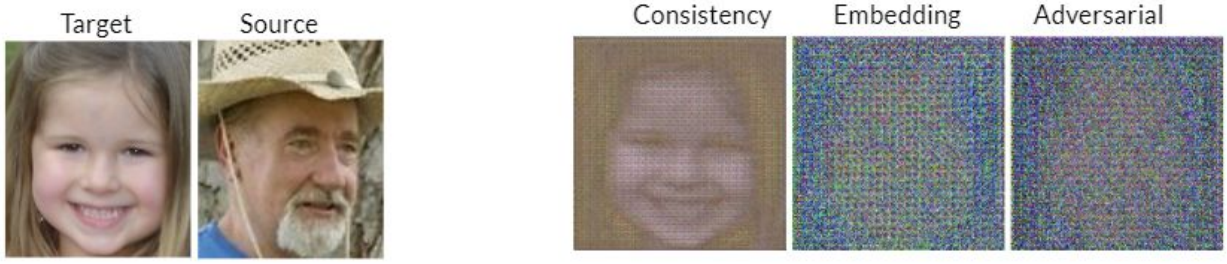
$(\alpha, \beta, \gamma) = (20, 0.1, 1)$

$(\alpha, \beta, \gamma) = (1000, 0.1, 1)$

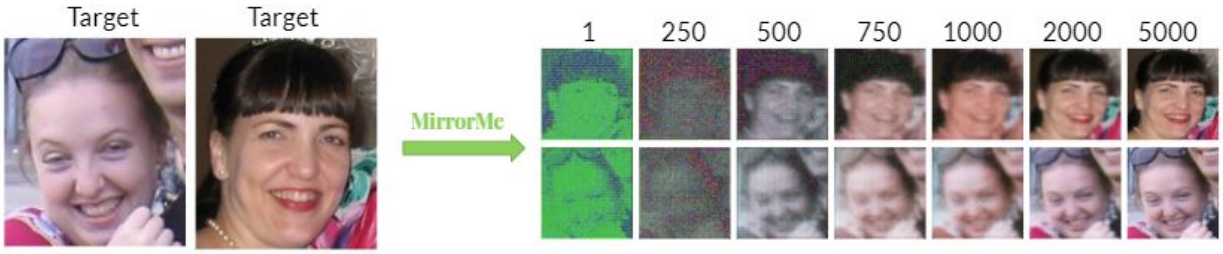
Figure 13. Loss plots for final model with various penalty weights

Qualitative Results

To investigate the effect of each loss metric on the final output, we ran the model three times, each time leaving two of the three loss weights as zero. The results below show, as expected, the consistency loss has the greatest effect on whether the output resembles the target image. The images were different enough to conclude that each loss makes a unique contribution.



Next are the results for our initial model, wherein we had 6 UNet layers and concatenated the facial expression embedding at the bottom of the U. We see that the reconstruction of the target images improves per epoch, but the expression does not change.



We tried concatenating the source and target images at the top left of the U along the channels:

Increasing the depth of the UNet to 14:

Neither approach had adequate results, and these data do not reflect the trend of low losses we observed.

Although our baseline was able to project the face of the source onto the target, it was unable to maintain the identity of the target (see “Baseline” section). MirrorMe included a consistency loss to maintain the style of the target and only vary the expression. Although it succeeded in

outputting the same person/character, it was unable to project the source expression onto the target.

Discussion and Learnings

Our final model did not perform successfully; there was no change in expression

One hypothesis was the adversarial and consistency losses overshadowed the embedding loss. We tried numerous ways to make the embedding more informative, such as increasing α and using other distance loss functions (cosine), but saw no improvement. Increasing the embedding dimension was infeasible due to lack of time and computational resources.

The failure to transfer expressions suggest that the embeddings provided inadequate information for this task, and were filtered out as random noise.

Embedding	Distance		
	ℓ_1	ℓ_2	Cosine
FACSNet-CL-F	47.1	47.1	40.7
FACSNet-CL-P	45.3	44.2	48.3
AFFNet-CL-F	49.0	47.7	49.0
AFFNet-CL-P	52.4	51.6	53.3
AFFNet-TL	-	49.6	-
FECNet-16d	-	81.8	-

Table 2: Triplet prediction accuracy on the FEC test set.

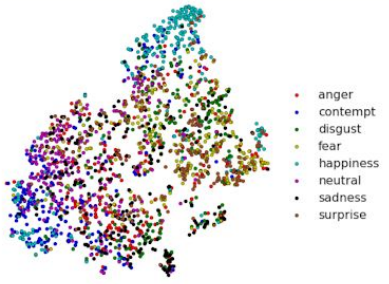


Figure 6: 2D visualization of the FECNet-16d embedding space using t-SNE [3].

[1]

The paper achieved 82% triplet accuracy. However, due to the lack of published code, only a pre-trained model with 64% accuracy was accessible. It is likely that these vectors do not describe an expression sufficiently, given that accuracy is somewhat low even for the simpler task of expression similarity comparison.

Another consideration is a visualization of the embedding space provided in the paper. Here, there is significant overlap between expressions of different emotions. This can suggest an inability to capture distinguishing details in expressions.

Throughout this project, we learned the importance of sufficient performance of each submodel, and the benefits of a modularized approach. Although we ran out of time/resources to improve the FECNet, we successfully pinpointed the main issue and would have been well-positioned to address it.

The arbitrary nature of the loss was also highlighted. A potential solution is normalizing the loss values to ensure certain losses are not weighed higher than others.

We also learned how parts of our network might be working against each other, eg. The importance of embedding loss is being diminished in favour of adversarial loss, and the Translator is learning to “cheat” the Discriminator by returning the same image. One potential

solution is training the Translator against different losses at different times, ie. against embedding loss first, then against adversarial/consistency loss.

For future projects, we would also like to explore more complex architectures than the UNet, such as StyleGAN[2], that are better equipped to distinguish face details hierarchically.

Ethical Framework

The premise of this project involves several stakeholders, including but not limited to:

- Celebrities, famous people
- Creative artists
- General populace
- (MirrorMe) Users
- Us (MirrorMe creators)

Had our model been successful, the following ethical implications would have followed.

Autonomy, Nonmaleficence

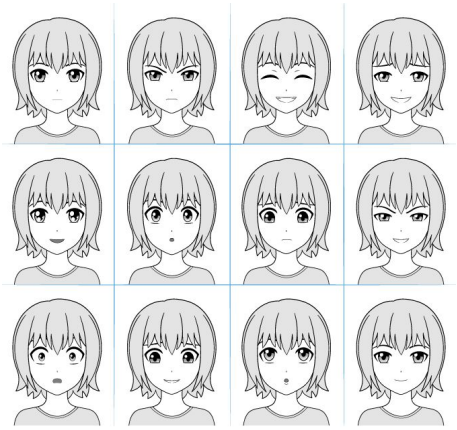
The ethical implications of MirrorMe lie at the intersection of facial recognition and Deepfake. These do not require the cooperation of the subject, and their main concerns are privacy, data protection, impersonation, and misrepresentation[5], [6]. MirrorMe is similar; it is ultimately up to the person using the model to decide if they will respect autonomy.

Misrepresentation is the only valid concern for MirrorMe, since the model does not apply to impersonation. **Famous people**, and the **general population** to an extent, would be most impacted. Additionally, inappropriate usage could result in backlash against the **creators** and **users**. Note, disrespect of Autonomy/Nonmaleficence is already enabled by technologies; MirrorMe does not exceed the degree of violation that is already perpetuated.

Beneficence

The use of MirrorMe for **creative artists** could reap many benefits, from more precise recreation of human expression in animation, to the projection of multiple facial expressions on the same character.

As a **general population** entertainment tool, MirrorMe could take forms from Snapchat filters to fanart tools.



Justice

The benefits, risks, and costs of using MirrorMe and being subject to the model is fairly distributed across the **general population**. Benefits are higher for **creative artists**, and risks, costs are higher for **celebrities**, **creators**, **users**. A case in which justice is not distributed is if the model simply does not work on that subset of the population. For example, if MirrorMe was less effective at transferring expressions between animated characters and East Asians.

Permissions

- permission to post video: **wait till see video**
- permission to post final report: **yes**
- permission to post source code: **yes**

References

[1] R. Vemulapalli and A. Agarwala, “A Compact Embedding for Facial Expression Similarity,” Nov. 2018, Accessed: Oct. 28, 2020. [Online]. Available: https://arxiv.org/abs/1811.11283v2.

[2] C. Yang and S.-N. Lim, “Unconstrained Facial Expression Transfer using Style-based Generator,” Dec. 2019, Accessed: Oct. 28, 2020. [Online]. Available: https://arxiv.org/abs/1912.06253v1.

[3] “Google facial expression comparison dataset,” *Google Research*. https://research.google/tools/datasets/google-facial-expression/ (accessed Oct. 28, 2020).

[4] Y. Zheng *et al.*, “Cartoon Face Recognition: A Benchmark Dataset,” Jul. 2019, Accessed: Oct. 28, 2020. [Online]. Available: https://arxiv.org/abs/1907.13394v3.

[5] “Facial recognition system,” *Wikipedia*. Oct. 26, 2020, Accessed: Oct. 28, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Facial_recognition_system&oldid=985456953.

[6] “Deepfake,” *Wikipedia*. Oct. 28, 2020, Accessed: Oct. 28, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Deepfake&oldid=985952515.