Facial Expression Manipulation with Conditional Diffusion Model

Yihe Tang
Department of Computer Science
Stanford University
yihetang@stanford.edu

Wanyue Zhai
Department of Computer Science
Stanford University
wzhai702@stanford.edu

Bihan Liu
Department of Computer Science
Stanford University
bihanliu@stanford.edu

Abstract

In this project, we tackle the task of manipulating facial expressions on face images utilizing conditional diffusion models. Previous works have used GAN-based models [3, 4] to get satisfiable performance on similar tasks, while few have explored using diffusion models, the rising architecture in generative tasks, specifically on this task. Our design of adding an extra semantic encoder is inspired by DiffAE[9] by Preechakul et al., which nevertheless only examines manipulating degrees of and interpolating image attributes. Focusing on our task, we propose using a semantic-rich encoder, such as the bottleneck layer of a pre-trained VAE on the dataset, along with target emotion as the diffusion model condition. In our experiments, the VAE-based semantic encoder outperformed the baseline with only emotion label embedding and the model with CNN classifier-based model in terms of good consistency in facial identity before and after manipulation as well as noticeable emotion manipulation, quantified by F-1 score on generated results.

1 Introduction

Bell's palsy is a non-progressive neurological disorder of the facial nerve. It is estimated to affect 25-35 per 100,000 people in the United States.[6] Motivated by the desire to help those patients and even everyone show their emotions, we proposed building an emotion generation network that utilizes the superior generation capability of diffusion models to assist everyone in expressing their feelings. We will tackle the facial expression generation task, where the model takes a human face picture and a target emotion such as happiness or anger as input. Our model will generate a picture of the same person with facial expressions corresponding to the desired emotion. We foresee a wide range of possible applications for this problem, including constructing emotion-rich photos for facial paralysis patients and photo editing applications to retouch dissatisfied emotions. [1]

[1] Our code is available on https://github.com/TangYihe/CS230

CS230: Deep Learning, Fall 2022, Stanford University, CA.

2 Related Work

Emotion generation is a topic that has recently received much attention, and researchers have tried to solve the problem using different methods. One of the most well-known emotion generation models is the StarGAN [3, 4] by Choi et al. They used a discriminator that distinguishes real and fake images and a generator that produces a fake image using an image and target domain label. Such architecture allows multi-domain image-to-image translation using a single model.

But recent advances in the diffusion model call attention to new architecture to tackle the same problem. The Denoising Diffusion Probabilistic Model(DDPM) [11] uses a Markovian diffusion process that allows high quality image generatino without adversarial training. DiffAE [9] incorporated the concept of a variation of DDPM, DDIM, and used semantic encoding and stochastic encoding to condition the input image to provide semantically meaningful inputs and allow for high-quality reconstruction. However, while they provide a wide range of applications, they only provide attribute manipulation, for example, manipulating the hair to be more wavey or less wavey, but it cannot directly manipulate to a new facial expression, which will be discussed in this paper.

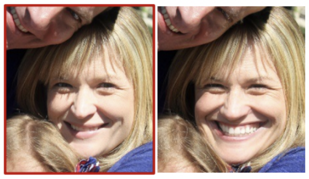


Figure 1: Smiling attribute manipulation with DiffAE.[9]

Better feature extraction and image reconstruction models like Variational Autoencoders(VAEs) are also powerful tools for learning the underlying image structure. VAEs are composed of an encoder, which maps input data to a latent representation, and a decoder, which maps the latent representation back to the original data space. The special characteristic that VAE supervises the posterior distribution of the latent encoding allows the latent feature extracted follow a nicer (in practice, mostly Gaussian) distribution and thus more generalizable when being used as pretraining technique for downstream tasks.[13]

3 Dataset

We trained, evaluated, and tested the baseline and our novel models on the FER-2013 dataset. The FER-2013 dataset is one of the most used datasets for facial expression research. It consists of grayscale images of emotional faces and labels the facial expressions into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set contains 28,709 examples, and the test set contains 3,589 examples.[5]

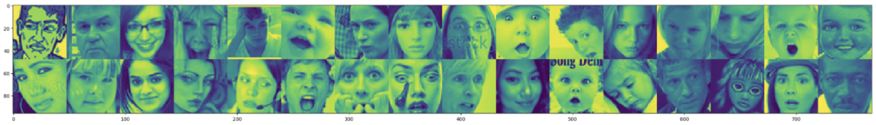


Figure 2: Sample FER-2013 dataset (normalized).[1]

Although using such a dataset can assist in model conditioning on emotion as it has clearly used emotion as labels, the potential limitations of the FER-2013 dataset are obvious. Firstly, the size of the dataset not big enough to account for the level of variations in the data. As we have read through some previous works on the task with GANs, most work used "cleaner" datasets collected under labotory settings. Secondly, the quality of the images is substandard since there are sometimes artifact images that are not from "real" people and only 1 color channel. These limitations require efforts in data pre-processing and potential catches in producing ideal results. However, we still chose to use FER as our dataset since the images all have clear facial expression, instead of having majority of the dataset with neutral faces, so it will be more demonstrative for our intended purpose.

4 Methods

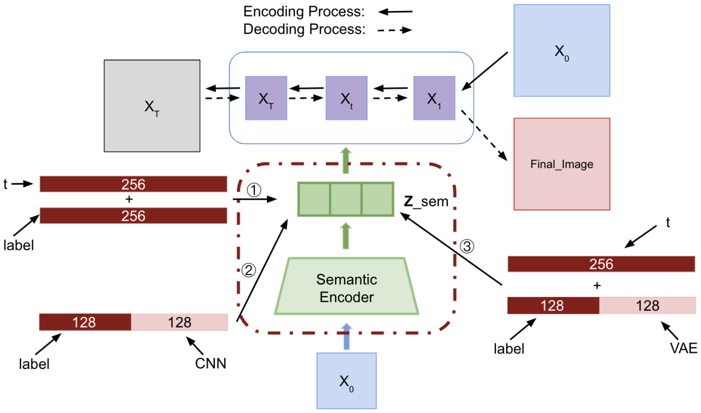


Figure 3: Model Architecture

4.1 Conditional Diffusion Model

To tackle the task of facial expression generation, we want to make use of the diffusion models (specifically DDPM) ’s 3 good capability of reconstructing and manipulating images. A vanilla diffusion model takes an image as input x_0 and iteratively adds Gaussian noise to it for some configurable number of timestamps T to generate a sequence of noised inputs $x_1, ..., x_T$. Ideally, we want x_T to follow a Gaussian distribution, i.e., behave as random noise. The model has a denoising network that predicts the noise being added at a given timestamp given the noised result, i.e., $p(x_{t-1}|x_t)$. In general, the denoising network is trained with MSE loss between the actual noise being added and the predicted noise. With this, at inference time, the user can start at timestamp T and sample a random noise, iteratively use the denoising network to predict the noise being added, and remove the noise until reaching the first timestamp, when the output is an image generated by the diffusion network.

The conditional diffusion model has the same general architecture, except for the fact that the denoising network takes an extra input "condition" in addition to the noised input x_t. In our project, we are interested in investigating how the different designs of the condition that got passed in could help guide the model to generate the emotional expression manipulated face that we are looking for. We will discuss this in the following subsections, and for all models mentioned, the training and inference procedure is as described above unless otherwise specified.

4.2 Baseline

In our baseline model, the denoising network condition contains an embedding of the timestamp information and the semantic encoding for the output by projecting the emotion label to 256-dim with an embedding layer. The additional label embedding aims to guide the denoising process by emphasizing the target facial expression we want in the output image.

The training process is supervised by MSE loss between predicted and actual noise, as described above. Specifically, during training time, the label getting passed into the semantic embedding layer is the facial expression label of the input image. During inference time, we will pass in the label for our desired emotion. This model supports both random sampling and input manipulation, where the only difference is the inference input being random sampled noise or forward (adding noise) result of the input image.

4.3 Semantic Rich Conditions

As we will describe in more detail in the next section, due to the fact that the denoise condition only contains timestamp and target label information, the previous baseline model performs well in generating images with target emotion but performs poorly in ensuring the generated face is still the image of the same person as the input image, which is an essential part in our task of manipulating facial expressions. Therefore, based on this observation, we propose our modified model, which incorporates semantic-rich encoding of the input in denoise condition, which guides the model only to modify the facial expression and keep the identity of the face in generated output unchanged.

Inspired by DiffAE, in which the authors used a CNN network to obtain semantic encodings for DDIM, our first design also used a CNN for semantic extraction.[9] Specifically, we first train a facial expression classifier with the standard structure of CNN followed by MLP projection on the FER2013 dataset. Then we use the value of the second last MLP layer (the last is the output layer with the dimension of num_emotion_classes) when passing in the input image as the source of semantic encoding. The intuition behind this design is that the later layer in neural networks will capture more global features of the input, and since the training is supervised with emotion classification results, the extracted features would be relevant to the facial expression-related features of the image (for example, whether the mouth is smiling).

However, a foreseeable place for improvement in the previous design is that emotion recognition is less relevant to some face characteristics that we still want to contain information about, such as the hairstyle or age of the person. Therefore, we further improve the design by pretraining a VAE on FER2013 and using the latent encoding given by the VAE as a source of semantic embedding. We can tell from Figure 4 that the VAE reconstruction results not only recover the general face characteristics such as gender, face angle, and hairstyle but also maintains the facial expression information, even not fully accurate. This provides strong support that the latent variable, being the information bottleneck in the VAE, must contain this semantic information, thus making our design reasonable.

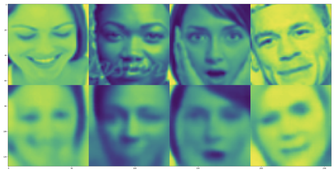
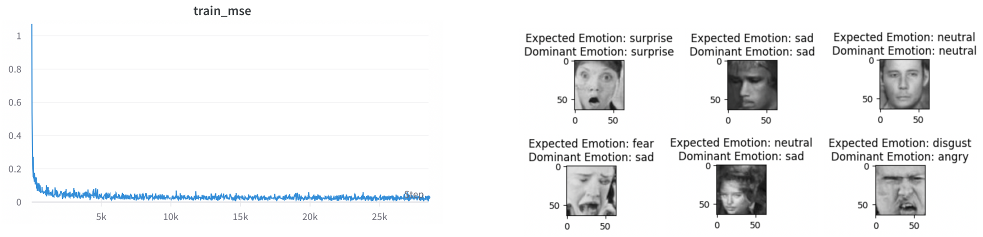


Figure 4: VAE reconstruction results

For the two designs mentioned above, we have a fully connected layer in our model that maps the encoding from either CNN or VAE to a 128-dim semantic embedding of the input. Then we concatenate it with the label embedding of the target expression and add them together with the timestamp embedding to become the denoising condition.3 The training procedure is similar to the baseline, and at inference time, we pass in the target label embedding and input semantic encoding as the denoise condition.

5 Results and Discussions



(a) Training MSE plot

(b) Baseline classified Results

Figure 5: Baseline Sample Results

Fig 6 shows our example generated images from the current network with input being 'neutral.' We also evaluated the generated images by applying an emotion classifier initially developed by DeepFace[10]. Fig 5b contains some example classified results. Notice that we have generated images with our expected emotions on the first rows but wrongly classified images that result from the imperfect generation. Table 1 shows the quantitative metric results per emotion label. It interprets the whole image as a face and performs emotion classification. Combining the result from quantitative analysis and our qualitative observation from the sample images generated by the three models, we can observe that the baseline model using label embedding only produces the best result on emotion classification, but it generates emotion from random faces. This results from the label embedding as the single condition, and it does not receive enough information about the target face. Our first semantic-rich embedding CNN captures the features of the targeted faces. But it produces worse results on emotion classification. The low performance on emotion classification is related to the performance of CNN as an emotion classifier. Due to the high variance in the distribution of our dataset and the lack of data, the CNN emotion classifier does not have a high performance, which affects its ability to provide distinctions in different emotions.

Our proposed VAE embedding performs relatively well on generating the correct emotion, and the emotions were generated on the same person. Given that our task aims to manipulate facial expressions from a given face image, VAE achieves the best performance. We believe that the reason why VAE achieves a lower emotion classification score than the baseline model is that the baseline embedding (label embedding) focuses only on achieving the desired emotion, but the VAE embedding takes into account the facial features as well as the emotion encoding on the faces.

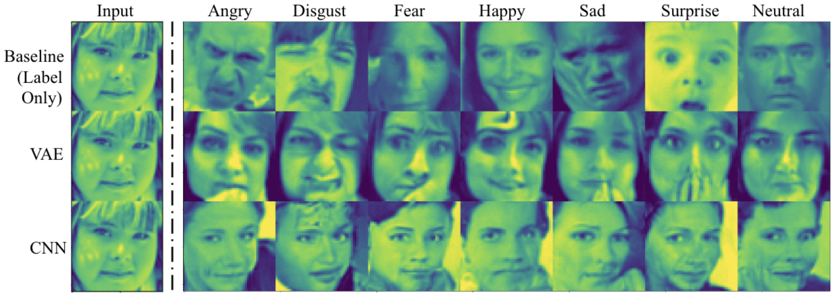


Figure 6: New Architecture Sample Results

Emotion	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Baseline(Label)	0.51	0.45	0.45	0.88	0.46	0.94	0.62
CNN	0.18	0.05	0.14	0.20	0.16	0.19	0.17
VAEs	0.33	0.15	0.26	0.56	0.32	0.42	0.30

Table 1: F-1 score of generated facial expressions using DeepFace

6 Novelty Impact

Our project can be considered novel for applying a new model to an application and modifying the structure of an existing model. Since diffusion models have only recently come into the public eye, based on our research on emotion generation applications, there is seldom any research that has been done on different emotion generation using the conditional diffusion model. We could only find the task using models like conditional GANs[14], so we built the entire baseline using the model[12], and fine-tuning it to these new tasks gets the very first sets of outputs.

For our model's new architecture, on the structure side, we explored and built new semantic encoder architectures applying ideas from CNN, VAEs, and training strategies specific for emotion encoding to build a better feature extraction model, which was not done in any original research.

7 Future Work

Since the quality current dataset could be improved, we propose the next step is to research finding functional potential better datasets. For example, there's a potential of utilizing the idea of transfer learning to train semantic encoding from a specially collected, facial expression rich dataset such as FER or other labotory setting ones, then migrate to a DDPM pretrained on a larger but less emotionally rich human face dataset such as CelebA. This requires a few more designs on model supervision strategies, but our work has shown that diffusion models paired with VAE pretrained semantic encoders has great potential of fulfilling this task.

There is also room for doing experiments using Bell's Palsy Patients data so that we can apply the model to real-life healthcare applications.

8 Contributions

Yihe built and trained the conditional diffusion model for both the baseline and the new architecture and maintained the GitHub repo. Wanyue built the evaluation pipeline, and performed quantitative and qualitative evaluations for our models. Bihan built the classification pipeline and trained CNN and VAEs on the FER dataset.

9 Jointly Conducted Project

This project shares a few code bases with Yihe Tang's final project, explicitly the data processing pipeline for FER dataset and the CNN model implementation. However, all the experiment conduction and diffusion, CNN, VAE model trainings are work for this project only.

References

[1] Barsoum et al. (2016). in Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution: FER+ (Face Expression Recognition Plus dataset). arXiv preprint arXiv:1608.01041.

[2] Capelle, T. (2022). Diffusion-Models-pytorch. GitHub. Retrieve from: https://github.com/tcapelle/Diffusion-Models-pytorch

[3] Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8789-8797).

[4] Choi, Y., Uh, Y., Yoo, J., Ha, J. W. (2020). Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8188-8197).

[5] Gilden, D. H. (2004). Bell's palsy. New England Journal of Medicine, 351(13), 1323-1331.

[6] Hauser, W. A., Karnes, W. E., Annis, J., Kurland, L. T. (1971, April). Incidence and prognosis of Bell's palsy in the population of Rochester, Minnesota. In Mayo Clinic Proceedings (Vol. 46, No. 4, pp. 258-264).

[7] Ho, J., Jain, A., Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33, 6840-6851.

[8] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[9] Preechakul, K., Chatthee, N., Wizadwongsa, S., Suwajanakorn, S. (2022). Diffusion autoencoders: Toward a meaningful and decodable representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10619-10629).

[10] Serengil, S. (2018). Deep Face Recognition with Keras. Retrieve from: https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/

[11] Song, J., Meng, C., Ermon, S. (2020). Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502.

[12] Song, L., Lu, Z., He, R., Sun, Z., Tan, T. (2018, October). Geometry guided adversarial facial expression synthesis. In Proceedings of the 26th ACM international conference on Multimedia (pp. 627-635).

[13] Subramanian, A.K. (2020). PyTorch-VAE. GitHub. Retrieve from: https://github.com/AntixK/PyTorch-VAE.

[14] Xiao, Z., Morris, T. (2018). Using Generative Adversarial Networks (GANs) for Facial Expression Generation.

[15] Yeh, R., Liu, Z., Goldman, D. B., Agarwala, A. (2016). Semantic facial expression editing using autoencoded flow. arXiv preprint arXiv:1611.09961.

Appendix

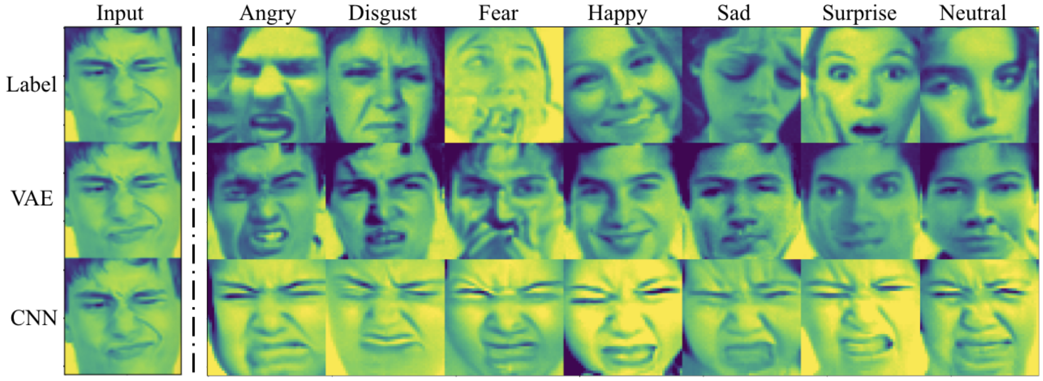


Figure 7: Sample Results from disgust input

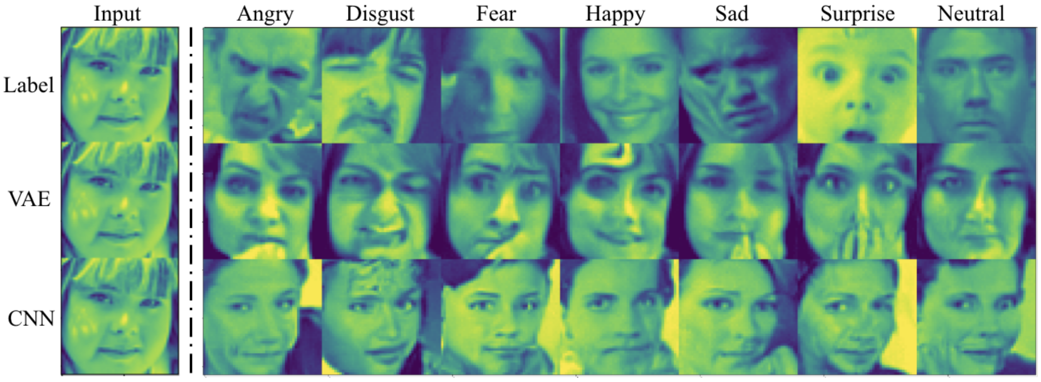


Figure 8: Sample Results from neutral input