Facial Expression Manipulation with Conditional Diffusion Model

Tianyi Cheng
Boston University
chengty@bu.edu

Haolin Ye
Boston University
haolinye@bu.edu

Runchuan Feng
Boston University
fengrc@bu.edu

Abstract

Facial expression manipulation has long been a significant area of research in the fields of computer vision and machine learning. Its potential applications are widespread, from entertainment to healthcare. Facial expression manipulation systems can generate realistic facial expressions given a source image and a target expression label. This project aims to develop such a system using diffusion models and U-Net. Diffusion models are generative models that learn the probability distribution of images and can be used to generate new images. The U-Net architecture is a deep neural network designed for image segmentation. By leveraging the recent advances in diffusion models and U-Net and using the CelebA dataset, the proposed facial expression manipulation system has the potential to generate high-quality images with conditional facial images.

1. Introduction

Facial expression manipulation has long been a significant area of research in the fields of computer vision and machine learning. Its potential applications are widespread, from entertainment to healthcare. Facial expression manipulation systems can generate realistic facial expressions given a source image and a target expression label. This project aims to develop such a system using diffusion models and U-Net. Diffusion models are generative models that learn the probability distribution of images and can be used to generate new images. The U-Net architecture is a deep neural network designed for image segmentation. By leveraging the recent advances in diffusion models and U-Net and using the CelebA dataset, the proposed facial expression manipulation system has the potential to generate high-quality images with conditional facial images.

2. Datasets

We plan to use the CelebA dataset. The CelebA dataset is a large-scale face attributes dataset containing more than 200,000 celebrity images, each annotated with 40 attribute annotations. The attributes cover a wide range of characteristics such as facial landmarks, facial expressions, hair color, and accessories. The dataset is commonly used in computer vision research, particularly in tasks related to facial recognition, facial expression analysis, and image synthesis. The images in the dataset have varying resolutions, aspect ratios, and facial poses, making it suitable for training deep learning models that can handle complex and diverse image inputs.

3. Approach

3.1. Diffusion models

Diffusion models are mathematical frameworks used to simulate the diffusion of substances in physical systems. They describe the movement of molecules in a system based on their concentration gradient, and have applications in fields such as physics, chemistry, and biology. In the context of the conditional diffusion model, the diffusion process is used to generate synthetic images by gradually introducing structure to a noise image.

3.2. Forward and reverse processes

The forward process of the diffusion model is defined by the formula $x_t = \sqrt{\alpha_t} x_{t-1} - \sqrt{1-\alpha_t} z_1$ where z_1 is Gaussian noise and $z_1 \sim N(0,1)$. This formula describes how the image x evolves over time t, given its previous state x(t-1) and a noise term z_1. The noise level $_t$ determines the strength of the noise that will be added to the image in that iteration.

The reverse process of the diffusion model is used to sample images from the learned distribution. Starting with the final image x_T, the reverse process generates intermediate images x_0 for t=T-1,...,0, by applying the formula $x_0 = \frac{x_t - \sqrt{1-\overline{\alpha_t}}\epsilon}{\sqrt{\overline{\alpha_t}}}$ where ϵ is obtained using Unet. This formula describes how the image x is generated backwards in time, given its future state x_{t+1} and a noise term .

3.3. UNet classifier

In the context of the conditional diffusion model, a UNet classifier can be used to generate the noise image . The

UNet takes the target image x_T and a set of conditioning variables as input, and outputs the noise image . The conditioning variables can be used to control the style and content of the generated image, and can be obtained from various sources, such as segmentation masks or feature maps.

The UNet classifier is a type of convolutional neural network that is commonly used for image segmentation tasks. It consists of an encoder network that downsamples the input image and extracts features, and a decoder network that upsamples the features and generates a segmentation mask. In the context of the conditional diffusion model, the UNet is modified to generate the noise image , by replacing the segmentation mask output with a noise image output.

3.4. Training and loss function

During training, the diffusion process is iteratively applied to the noise image, with the noise level $_t$ increasing at each iteration. The loss function used to train the model is based on the negative log-likelihood of the generated images under the learned distribution, and can be augmented with additional terms to capture desired characteristics of the generated images.

For example, additional terms can be added to the loss function to penalize images that have high levels of noise or that do not meet certain stylistic or content-related constraints. These constraints can be specified using the conditioning variables, such as specifying the desired color or texture of the generated image.

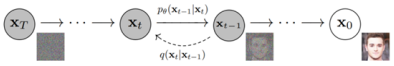

Figure 1. Diffusion model illustration.

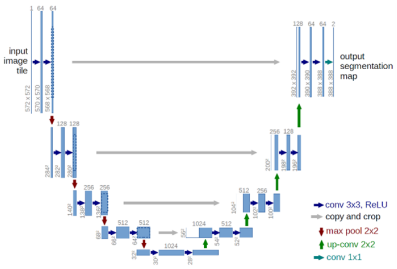

Figure 2. U-Net architecture.

4. Coding Explanation

The provided code implements a UNet-based diffusion model for modifying the attributes of a given image. The code is designed to work with the CelebA dataset, which contains facial images with 40 different attributes such as 'Smiling', 'Gender', 'Young', etc.

The code defines a PyTorch dataset class called 'CelebA-Dataset' that reads the CelebA dataset and loads images and their corresponding attribute labels. The attribute labels are binary (0 or 1) and represent whether an attribute is present or not in the given image.

The UNet-based diffusion model is defined in the 'UNet' class. The model takes an input image and a target attribute vector as input and produces a modified image that has the desired attribute. The UNet architecture is composed of a series of down-sampling and up-sampling blocks, each consisting of a residual block that includes a GroupNorm layer, a convolution layer, and a non-linear activation function.

The code also defines a 'Block' class that implements a residual block used by the UNet architecture. The 'Down' and 'Up' classes implement the down-sampling and up-sampling blocks, respectively.

Finally, the code provides a training loop that trains the diffusion model on the CelebA dataset. The model is trained to modify the given image such that it has the desired attributes. During training, the diffusion model is trained using an algorithm, which involves adding noise to the input image and updating the image through a series of time-steps until the desired attributes are achieved. The training process is guided by a contrastive loss function that encourages the modified image to be similar to the original image. The training loop also logs the training progress using the 'tqdm' library and saves the model checkpoint at regular intervals.

5. Results

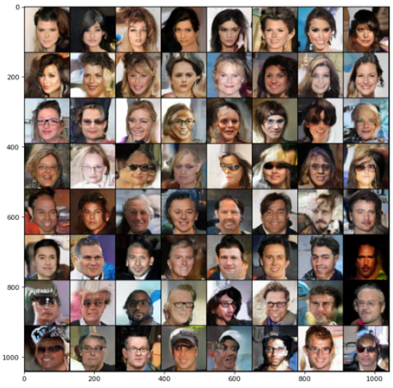

Figure 3. Results of the facial expression manipulation. Lines 1 and 2: Female without glasses. Lines 3 and 4: Female with glasses.

Figure 4. Results of the facial expression manipulation. Lines 5 and 6: Male without glasses. Lines 7 and 8: Male with glasses.

6. Discussion

6.1. Use a different conditioning mechanism

One possible modification to the conditioning mechanism is to use multiple sets of conditioning variables that correspond to different regions or features of the image. This can allow for more fine-grained control over the style and content of the generated images.

In the context of image segmentation, for example, a UNet classifier can be used to generate a binary mask that indicates which regions of the image correspond to different objects or features. This mask can then be used to condition the diffusion process, such that different sets of conditioning variables are used for different regions of the image. By allowing each region to have its own set of conditioning variables, the model can generate more realistic and diverse images with more varied styles and features.

Alternatively, other classifiers can be used to generate the conditioning variables. For example, a convolutional neural network (CNN) can be trained to extract features from the image, and these features can then be used to condition the diffusion process. The use of different classifiers for conditioning can result in different levels of control over the generated images, and the choice of classifier will depend on the specific requirements of the task at hand.

6.2. Modify the loss function

The loss function used to train a conditional diffusion model measures the difference between the generated images and the real images in the training dataset. By modifying the loss function, it is possible to better capture the desired characteristics of the generated images.

For example, adding a term to the loss function that penalizes images with high levels of noise can result in clearer and sharper images. Similarly, adding constraints to the loss function that require the generated images to meet certain stylistic or content-related criteria can result in images that are more consistent with the desired style or content.

6.3. Change the architecture

The architecture of a conditional diffusion model determines the way in which the image is generated. Changing the architecture can allow the model to capture more complex relationships between the different features of the image, resulting in higher-quality and more diverse images.

For example, adding more layers to the model can increase its capacity to capture complex nonlinear relationships between the features of the image. Similarly, using a different type of activation function can improve the model's ability to capture the nonlinearities in the data.

In conclusion, there are several different methods for modifying conditional diffusion models to improve their performance. By using a different conditioning mechanism, modifying the loss function, or changing the architecture, it is possible to generate higher-quality and more diverse images that better meet the desired style and content criteria. However, it is important to carefully consider the trade-offs between these different methods and to choose the approach that is best suited for the specific problem at hand.

7. Statement of Individual Contribution

In this project, our team members Haolin Ye, Runchuan Feng, and Tianyi Cheng have contributed significantly to the successful completion of the project. Each member has played a vital role in developing the Facial Expression Manipulation system using Conditional Diffusion Model and U-Net. The individual contributions of each team member are as follows:

7.1. Haolin Ye

1. Pre-processing and handling of the CelebA dataset, including the implementation of the `CelebADataset` class.

2. Collaborated with Tianyi Cheng on the development of the training loop, including the implementation of the contrastive loss function and logging the training progress.

3. Model evaluation and generation of visual results showcasing the performance of the facial expression manipulation system.

7.2. Runchuan Feng

1. In-depth analysis and understanding of the underlying mathematics and formulas involved in the diffusion model.

2. Assistance in the implementation of the diffusion model code, including the forward and reverse processes.

3. Writing the project report, including the introduction, approach, coding explanation, and results sections.

7.3. Tianyi Cheng

1. Literature review and research on diffusion models and U-Net architecture for facial expression manipulation.

2. Development of the UNet-based diffusion model code, including defining the `UNet`, `Block`, `Down`, and `Up` classes.

3. Collaborated with Haolin Ye on the development of the training loop and implementing the contrastive loss function.

Our team has worked collaboratively to achieve the project objectives, and each member has contributed their expertise to ensure the development of a high-quality facial expression manipulation system using Conditional Diffusion Model and U-Net.

References

[1] 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).

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

[3] 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).