In recent years, Generative Adversarial Networks (GANs) have become the state of the art in artificial image generation. We applied Neural Style Transfer, Image Segmentation, and GANs to a new application, altering facial expressions in photos. This generative task may have useful applications in special effects or artistic tooling. As input our model takes an image of a person smiling, and as output produces an image with a neutral facial expression.

**Problem Statement:** Is it possible to use neural networks to change a person’s facial expression?

### Custom Dataset
- For SA-GAN, used only JAFFE dataset
- JAFFE consists of 213 images of 10 distinct Japanese women
- Each makes 6 different facial expressions (anger, disgust, fear, happiness, sadness, and surprise)
- For Cycle-GAN, simplified to 2-class problem, padded dataset with faces from FEI Dataset
- Cycle-GAN training set had 446 images, each with 1 channel

### Approaches
- Experimented with various architectures and models
- **Style Transfer + Segmentation:** transplant face + smooth
- **Self-Attention C-GAN:** CGAN + module to model long-range spatial feature dependencies
- **Cycle GAN:** GAN transforms images between two classes, enforcing cycle consistency

### Experiments and Results
- **Self-Attention (SA-CGAN):** Completely learned image transformation, better quality than CGAN, but needs improvement
- **Mode collapse** frequent with SA-CGAN
- **Cycle-GAN:** Trained for 350 iterations, lr = 0.0005
  - Learned high quality transformation between Happy and Neutral expressions.
  - Micro and macro-level facial feature adjustments show clear model ability to interpret and manipulate expression

### Conclusions and Future Work
Both GANs were able to learn non-trivial expression changes on input images. Cycle-GAN cycle-consistent loss greatly increased performance, and produced more realistic results. This is likely because reducing the problem to two-class simplifies the prediction. For example, the model seemed to localize changes to the mouth area.

**Future Work:**
- Expand Cycle-Gan implementation to multiclass problem
- Experiment with Style-GAN architecture for better resolution

### References