# Conditional Generative Adversarial Networks for Flower Generation

Group 21: Zeeshan Dar, Sy Hai Dinh, Lars Olsson, Oleksii Prykhodko

October 14, 2018

Our project will be addressing the problem of generating images using conditional generative adversarial networks (CGANs). CGANs extend the GAN model by parameterizing the generated output. Our motivations are the interesting properties we can examine once we have the CGAN extension in place. Specifically, we can evaluate the models' understanding of the dataset through giving values in the parameter space not encountered while training.

We have chosen the VGG 102 Category Flower dataset. Care must be taken to choose a set of parameters matching with image characteristics, such that the networks can learn some function relating the two. We would also like for these parameters to have a number of unique values in the dataset, again to help the network learn the relevance of the parameters. Flowers exhibit both these qualities in their variable and visible features, such as color and number of petals.

Promising results of flowers being generated using GAN-based techniques exist, though not with a simple CGAN. Instead, these images are generated using a more complex text to image model. As our network is trying to achieve a similar goal, but does not have to learn to extract information from text, we can have a realistic expectation of our simpler approach working.

We haven't found code that addresses our problem in full. However, there is a good tutorial on how to generate a GAN. CGANs should be a relatively simple extension of this existing model. We aim to entirely rewrite the model in Keras, but if this proves infeasible, we will proceed with the existing Tensorflow implementation. We will then extend it to become a CGAN. Finally, we will tune the hyperparameters to maximize performance with our specific dataset.

There is an abundance of papers relevant to both GANs and CGANs. We will keep our reading to the classic end of literature - the original GAN paper as well as the original CGAN paper.

GANs can be tricky to evaluate. As a quantitative measure of acceptance, we will use a pre-trained image classifier to give a probability measure. This will be accomplished through grouping classes into 'flower' and 'not flower', and then seeing which set contains the class provided by the classifier when given images drawn from another set of generated results.

We cannot be sure at this point in time whether our CGANs will be able to learn the relevance of the parameters given the relatively small data size, so the aforementioned GAN evaluation technique will be our primary acceptance criterion.

Our secondary goal is the CGAN learning to parametrize the image space. This will be evaluated through manual inspection of generated images. We will feed new tuples of parameters, where each parameter has been seen in training, but not in this combination. Then we will try to extend our parameter space to include values not seen in training, to some small degree. Finally, we will test values that lie far outside our training parameter space.

**Tentative timeline:**

1. Read the relevant papers.
2. Preprocess data by giving parameters to classes, and possibly removing classes where this mapping is unclear.
3. Design and train network architecture in Keras.
4. Evaluate results; iterate tuning hyper-parameters to improve results.
5. Write report and poster.