GENERATING STYLISED GAME ASSETS USING GENERATIVE ADVERSARIAL NETWORKS

Paulina Tao

Supervisors:
James Walker, Simon Demediuk
Motivation

- **Revolution Studios**: Tremendous effort to hand draw every scene
What is GAN?

- First introduced in paper *Generative Adversarial Networks* (Ian J. Goodfellow, 2014)
- "A generative adversarial network (GAN) is a class of machine learning frameworks...Two neural networks contest with each other in a game (in the form of a zero-sum game, where one agent's gain is another agent's loss)." ----Wikipedia
- The generator tries to minimise the following function while the discriminator tries to maximise it:

\[
\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log (1 - D_{\theta_d}(G_{\theta_g}(z))) \right]
\]

Discriminator output for real data x
Discriminator output for generated fake data G(z)
- **Generator**: Aim to fool the discriminator by creating fake images
- **Discriminator**: Aim to identify fake and real images
- **Result**: The discriminator ends up guessing (50:50) about whether an image is real/fake, and we have the ideal generated image.
What are we aiming for?

- **Short Answer**: To explore, understand, and pioneer the generation of game assets through the use of GAN Technology

Content Image

Style Image

Output
What are the subgoals?

Which GAN architecture to use for which content/style combination?

Do parameters actually change the styles or do they just reduce the image quality?

If this architecture is good at generating one style, does it imply that it is also good with the others?

Is there a correlation between the change of a parameter and a particular style?

What kind of style is this particular architecture good at?

What and why do hyperparameters change the output the way they do?
What is our plan?

- Step 1: Background research for GANs that look most promising (cycleGAN, AdaIN, Lapstyle, Deep Photo Style)
- Step 2: Run combinations between different content and style images with the GAN
- Step 3: Pick the GAN that is most capable of generating the appropriate content to the style images.
- Step 4: Sweep across all content/style combinations and find out the correlation between the hyperparameters for the GAN and the output generated.
Results for Texture Samples

Brick Content Image

Brick Style Image

Cycle GAN

AdaIN
Results for Texture Samples

Stone Content Image

Stone Style Image

Cycle GAN

AdaIN
Results for Texture Samples

Fabric Content

Cycle GAN

AdaIN

Fabric Style Image
Results for Cityscape Samples

Top: Content Image
Bottom: Style Image
Results for Cityscape Samples

Outputs

Cycle GAN

AdaIN
Results for Cityscape Samples

Outputs

Deep Photo Style

Lapstyle
Future Work

• Continue hyper-parameters sweeping across all combinations of GAN type and content/style inputs
  ◦ Dive deep into the results to analyse what and why parameters cause such changes
• Produce a Journal article detailing the results
• Engage with industry to explore real-world applications, such as artist tool for automatic style driven outputs
Personal Experience

- Pros:
  - Supervisors are fantastic! :)
  - Learn a lot about GANs
  - Beneficial for Master's

- Con:
  - Research GitHub repositories are not meant for normal users…
  - Wish I knew before:
    - How to implement a GAN architecture from scratch