Generating RPG Monsters using VAE

Tholkappian Chidambaram, Robert Maxton, Matthew Zhu, Dushyanth Giridhar, and Abraham Williams

Idea

Generate artistic renditions of RPG (role-playing game) monsters

Fundamentally artistic and alien - “errors” in other domains are “features” here

Tend to share qualities - though we’re excluding palette swaps!

Use VAE (Variational Autoencoder) to generate “similar” output

Variational Autoencoders

An autoencoder is a neural network that consists of an encoder function and a decoder that produces an (originally identical) reconstruction.

Encoder “squeezes” out irrelevant/unused dimensions; decoder in the process of training learns a general reconstructor function that can be repurposed for generation.

“Variational”: By using random sampling and measuring similarity, learns to interpolate between given data points probabilistically.

Training Dataset

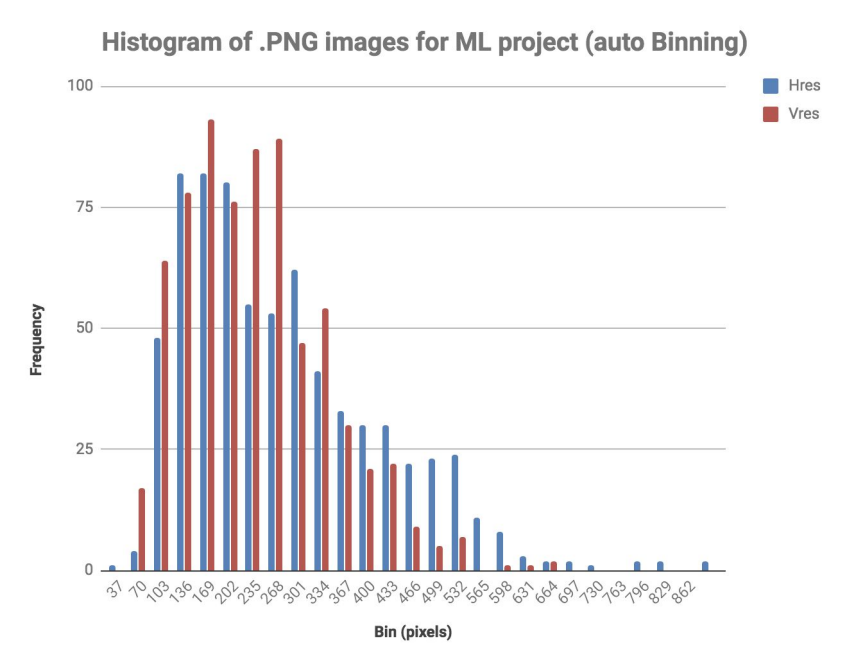
RPG monster assets from the Ækashics Librarium (http://www.akashics.moe/)

Initial dataset of 703 images, before pre-processing steps

Images were encoded in RGBA, where a is the opacity

Training Dataset (cont.)

Histogram of horizontal and vertical resolution of training images



Pre-processing

Removed the largest and smallest images

Larger images were downsampled to 320x480

Smaller images were padded with white space

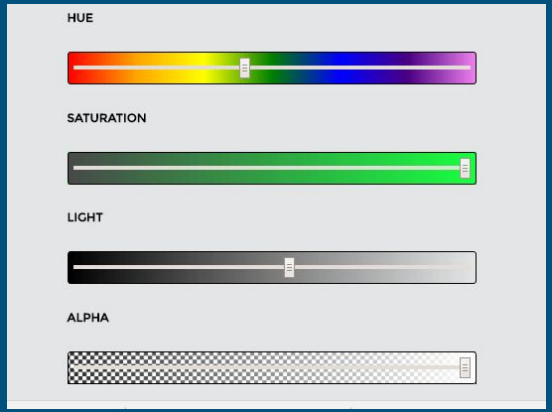
Completed training dataset consisted of images with 320x480 resolution

Pre-processing (cont.)

Converted the images from RGBA to HSLA (Hue-Saturation-Lightness-Alpha)

- Thus two dimensions (Hue-Saturation and Lightness-Alpha) can be achieved, prioritizing Hue and Lightness
- Otherwise in RGBA, we would have to choose to prioritize some colors over others

Entire training dataset is saved in
a CSV file, where each image is a row.



Model

Input layer is the pre-processed image

Arbitrary n hidden layers, which exponentially compresses the image at each step

Image is reduced to latent dimensions - minimum complexity representation

“Minimum” Complexity?

Most of the complexity of a monster is contained within its edge

We modeled the edge as a fractal (fractal dimension was a hyperparameter of the model; initial guess of 1.25)

Total LATENT_DIM is 2*(monster_height = image_width)^fractal_dim

Model (cont.)

The output of the encoder is then decoded for the hidden layers in reverse

Measured loss by log-likelihood and KL-divergence of the initial and final images, considered as probability distributions

This model creates images in HSLA, which would need to be converted back to RGBA to maintain consistency with the original training images

Summary

Retrieved RPG monster assets from Ækashics Librarium

Pre-processing to convert images to 320x480 resolution and HSLA format

VAE to create model that prioritizes Hue and Lightness and emphasizes the edge of a monster in estimating the complexity of the image

Google Cloud ML Engine, Keras, Tensorflow backend

References

https://tiao.io/post/tutorial-on-variational-autoencoders-with-a-concise-keras-implementation/

http://www.akashics.moe/