RYAN EMANUEL WRITE UP

INTRO

In the market for professional cinema cameras, Red Digital Cinema has a patent on in-camera RAW recording formats. RAW formats allow the same in-camera color correction controls for post production. From a technical perspective, this is useful if the incorrect camera settings were selected during recording or if creatively filmmakers want more options for color grading in post. The patent has forced manufacturers like Sony and Canon to record RAW outside the camera with external recorders. This makes the cameras bulkier and more expensive. Perhaps there is a Data Science solution to provide the same image transformations through an image processing pipeline.

The purpose of this project is to create an image processing pipeline to provide RAW functionality for encoded (non-RAW) images by preprocessing footage in NUKE with python scripts, sampling images by extracting RGB samples to generate a SQL database, and then feeding that data to an Artificial Neural Network model to approximate the image transformations. For this project, I’m focusing solely on white balance control. While the final model will be utilized as a post production plug-in, a web application will showcase the transform for visual analysis.

DATA

The dataset consists of just over 1 million data samples. The features comprised of an input white balance, RGB input sample triplet, output white balance, and RGB output sample triplet. The data was generated from 18 photos that I took on the Blackmagic Pocket 4k under daylight spectrum light. The photos had varying exposure of a chip chart with 700 swatches. I used Nuke’s python scripting, along with Google Colab, to permute all possible color temperature transforms between 9 common color temperatures, and to extract the RGB information from each color chart. Nuke was necessary to produce EXR files from the RAW SDK to guarantee no unintentional image transformations while being able to convert images to arrays.

**TOOLS**

**Data Engineering**

NUKE

OpenEXR

Skimage

PIL

**Data Processing**

Sklearn

Numpy

Tensorflow

**Data Storage**

SQL

**Deployment**

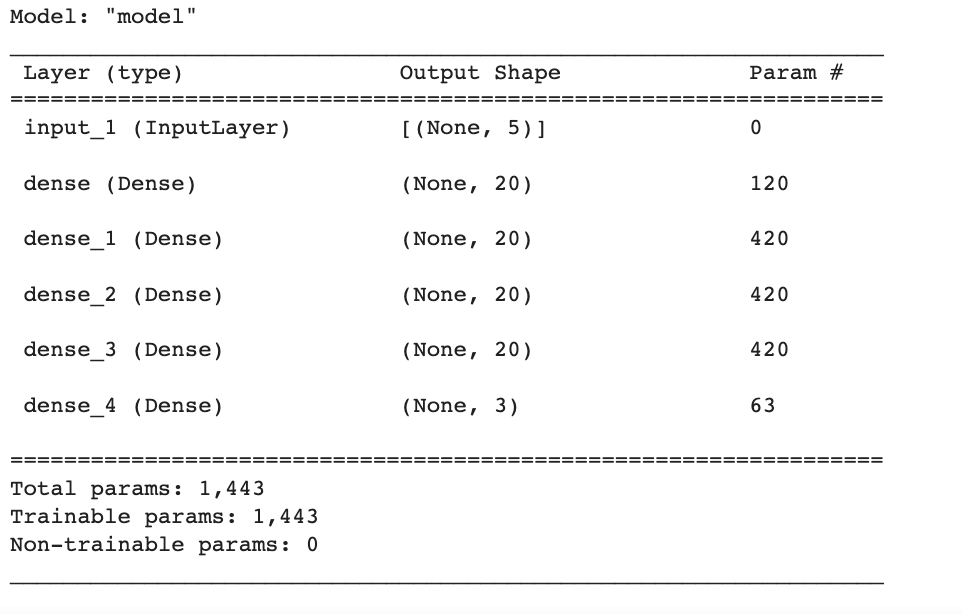
Colour

Flask

Gunicorn

**ALGORITHM**

The ANN model structure had 5 inputs: the input white balance, input RGB values and the output white balance.



The model needed a small amount of parameters for performance in post production plugins, but still had a MAE of 0.003 for the test set, which is great for a 0-1 domain.

**COMMUNICATION**

Here is a link (https://github.com/rzemanuel/Data-Engineering) to the github where the web application can be downloaded and run locally.