**Chrissy Smiley Capstone Technical Report**

**Image Classification with Convolutional Neural Network**

**Project Background**

The purpose of this project was to create a convolutional neural network image classifier to distinguish between two types of images: horror movie monsters and horror movie protagonists.

**Data Acquisition**

The data was acquired using the Fatkun Batch Download Image extension for Chrome as applied to Google Image Search results. Some of the image search terms used included “Horror Movie Monsters” and “Horror Movie Protagonists”. Each result was saved and manually assessed for compatibility with the project. The images that were retained were then filtered through to ensure there were no duplications. Ultimately there were 750 total images; 600 total training items belonging to two classes and 150 testing items belonging to two classes.

**Data Transformation**

The data was augmented as part of the model to provide additional training & testing images. The augmentation strategy included rescaling, shearing, zooming, and horizontal flips. These images were generated indefinitely during the fitting process using a data generator.

**Model**

Four versions of the convolutional neural network were created for this project: a base model, a model with dropout, a model with regularization, and a model with dropout and regularization.

The base model consists of three rounds of layers: a convolutional, activation, and pooling layer each – followed by a flattening layer, two dense layers, and a final activation layer.

The loss function used was “binary crossentropy”, the optimizer used was “Adam”, and the metric used was “accuracy.”

**Results**

* **Base Model** – acc: .90, val\_acc: .66
* **Dropout** - acc: .70, val\_acc: .56
* **Regularization** – acc: .86, val\_acc: .69
* **Dropout & Regularization -** acc: .78, val\_acc: .70

**Future Considerations**

Each model took over an hour to run on the available computer; in the future, processing on a more powerful computer would be desired.

Having the data loaded in a data generator format made running statistics on the model difficult. If this project were to be redone the data would be loaded in as an array providing for a classic train-test-split and available metrics.