**Predicting Captions for Images Taken By Blind Users**

**Abstract**

The goal of this project was to use a hybrid CNN/RNN deep learning model to classify image features and then predict captions for those images. I worked with data provided by [VizWiz.org](https://vizwiz.org/), who offered a challenge in 2020 to predict accurate captions on images taken by blind users. I used a convolutional neural network to identify image features, and a recurrent neural network (applying long short term memory) to predict word sequences, which were ultimately combined to predict captions on input images.

**Design**

This project is a non-competitive response to the [VizWiz Image Captioning Challenge](https://vizwiz.org/tasks-and-datasets/image-captioning/) and uses their challenge-curated data. The data itself consists of images, generated entirely by blind users, and five associated captions per image (generated by users with sight). Captioning images accurately via deep learning models could grant many people who are blind a new degree of independence when it comes to understanding the content of photographs they take. Beyond this scope, automated and accurate image captioning could be leveraged for a wide range of ‘watchdog’ applications.

**Data**

The dataset contains a total of 39,181 images and 195,905 captions. This data arrives split into training (23,431 images, 117,155 captions), validation (7750 images, 38,750 captions), and test (8,000 images, 40,000 captions) sets.

**Algorithms**

*Transfer Learning*

1. [Xception](https://iq.opengenus.org/xception-model/) was used to extract features from the training images

2. An unsuccessful attempt was made to extract features with [VGG16](https://www.mygreatlearning.com/blog/introduction-to-vgg16/).

*Natural Language Processing*

pandas, keras text preprocessing tools, and dictionary functions were used to create a language corpus and tokenize it.

*Neural Network Architecture – keras API*

***1. Feature Extraction –*** input > dropout > dense > output for add layer

***2. Sequence Processing –*** input > embedding > dropout > lstm > output for add layer

***3. Caption Prediction –*** add (outputs of feature extraction and sequence processing are combined) > dense > dense

**Accuracy** was used as a mathematical metric for overall model performance (best performance = 43% as of this writing)\*, as well as human judgment of the predicted caption for an input test image.

**Tools**

* NumPy, Pandas, Pickle, Pillow, and Xception were used for data cleaning and preparation
* Keras provided the deep learning API as well as corpus tokenization capacities
* Matplotlib was used to generate an image of the neural network architecture

**Communication**

In addition to the slides and presentation given April 20, 2022, this entire project in it’s current and future iterations will be available on my personal GitHub.