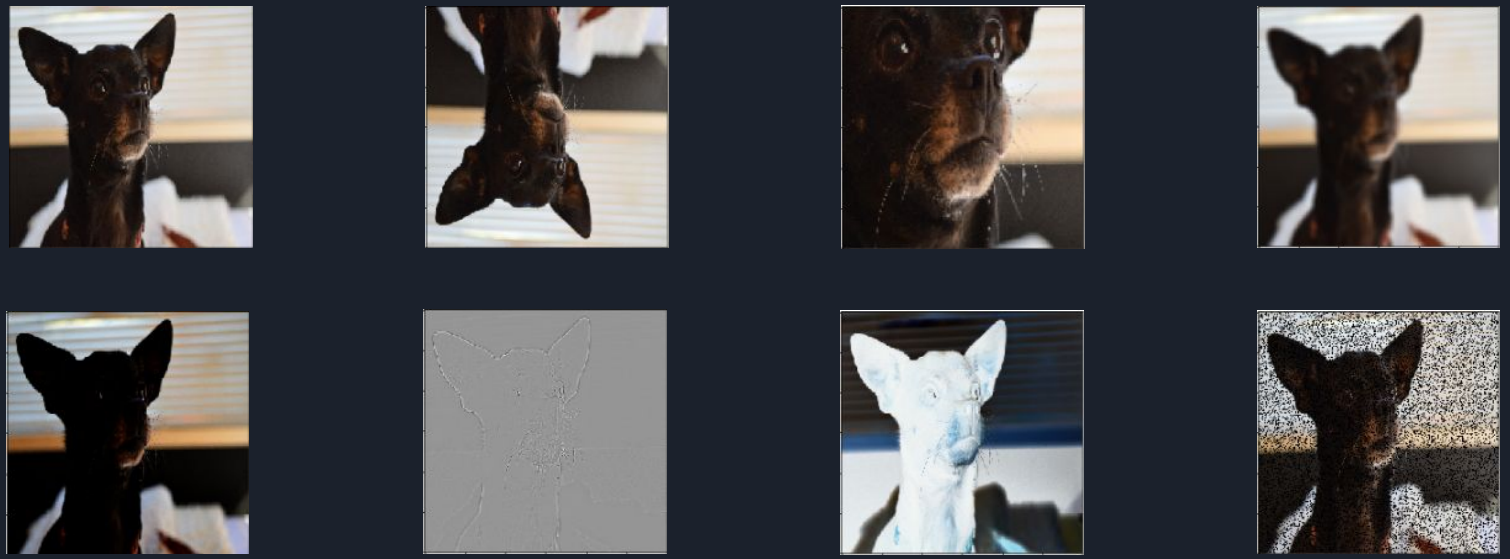
Using “imgaug” Python Library for Image Augmentation

Alex Williamson (alswilli@ucsc.edu)
UC Santa Cruz
CMPS 184 - Spring 2019

What is imgaug?

- Python library that allows for easy image augmentation
- Provides a wide variety of different augmentations that are commonly used when attempting to improve the task accuracy of AI being trained with images

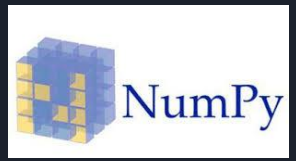


Summary of Tutorial

- **Goal of Tutorial**
  - Demonstrate how to use imgaug to improve classification accuracy of Deep Learning-based Image Classifier for two Chihuahuas
- **Tutorial Process**
  - Stores 26 images of a black Chihuahua (Scarlett) and 34 images of a white Chihuahua (Pistachio) for a total of 60 images
  - Two Experiments
    - Exp. 1 trains with 48 randomly selected non-augmented images and validated with the remaining 12 images
    - Exp. 2 trains with 48 randomly selected non-augmented images + 96 images from applying 2 augmentations to non-augmented images
      - Validation set gets same treatment (12 non-aug + 24 aug images)
    - Both models in the experiments are the same and trained for 50 epochs
- **Results of Tutorial**
  - Validation accuracy for Experiment 1 -> **63%**
  - Validation accuracy for Experiment 2 -> **100%**

Python Libraries Used

- **Main Libraries**
  - imgaug
  - tensorflow
  - keras
- **Additional Libraries**
  - sklearn
  - matplotlib
  - pandas
  - numpy
  - os, glob, pickle



Key Commands during Tutorial

- **img = image.load_img(imagePath)**
  - Uses keras.preprocessing to load an image from a path on computer
- **plt.imshow(img)**
  - Uses matplotlib.pyplot to display image to jupyter notebook cell
- **aug_img = aug.augment_image(img)**
  - Uses imgaug to apply an augmentation to img and save the new augmented image
- **seq = iaa.Sequential([aug1, aug2, aug3])**
  - Uses imgaug to allow multiple augmentations to be applied to one image at once
- **x_train, x_val, y_train, y_val = train_test_split(x, y, test_size=0.2)**
  - Uses sklearn.model_selection to randomly split image data into train and testing variables
- **model.compile(), model.fit(), and model.predict()**
  - Use keras to execute Deep Learning training process with train and testing variables

Link to Tutorial!

https://nbviewer.jupyter.org/github/alswilli/CMPS-184-ImgAug-Presentation/blob/master/CMPS%20184%20Presentation%20-%20imgaug%20Python%20library%20with%20Deep%20Learning.ipynb