Yoga pose classification

Problem Statement

Given a picture of a yoga pose, can a model be built within 1 year to return the name of the pose?

Computer vision and analysis has been applied to many different move and sport problems so far. Yoga is a natural step in that process. Two of the most cited reasons for not going to a yoga studio are travel constraints and embarrassment in not knowing how to do the poses properly. Removing the classroom and giving the new practitioner feedback and instruction in their home would answer both reasons. This analysis will be an initial step in that direction, identifying yoga poses from pictures. It will be considered successful if it delivers a model that can classify yoga pose images with 85% or better accuracy.

Data Wrangling

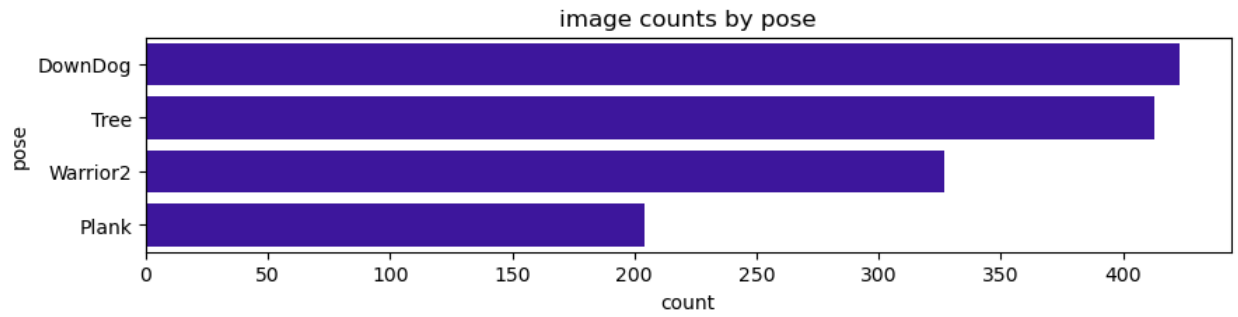
The data used for this analysis started with a set of images collected from a data set Niharika Pandit submitted to kaggle. The dataset contains 1,551 images of 5 yoga poses:

- Downdog
- Goddess
- Plank
- Tree
- Warrior2

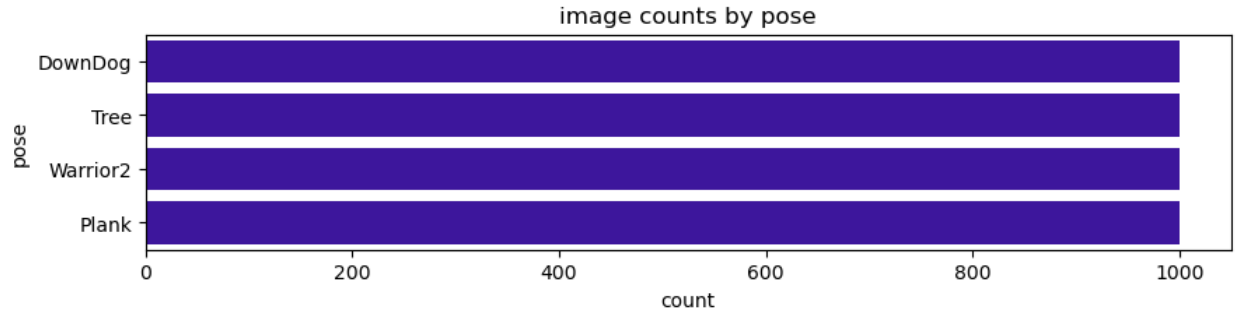
The train and test folders were combined and the contents of each pose folder reviewed. There were not enough images in the goddess folder to build proper train, test and validate splits, so this folder was removed. The remaining poses were supplemented with Google Image search results. Each image folder was manually reviewed to ensure all images were of the appropriate pose and contained one person performing the pose. For simplicity, images from all file formats other than jpeg were removed. The images were also validated for use in image classifiers and cleaned of duplicates. Some effort was made to ensure a good balance of body type, skin tone and sex were represented, but this was largely impossible as a true balance resulted in too few images to train models. Finally, all images were resized to 256x256 as this type of sizing ensures proper training for many types of classifiers.

Exploratory data analysis

After the images were cleaned, the image counts per folder were reviewed.



There was a strong class imbalance that would affect model training. Finding additional images that fit all of the criteria to download was becoming difficult. The data was split into train, validate and test sets. This was important to do here as altered images were going to be created and the validate and test sets would be less useful if they contained the altered images. Each pose had 100 random images moved to validation and 100 moved to test. With the remaining data deemed the training set, each pose was supplemented by selecting random images and applying random transformations until all poses had 1,000 images.



The final folder structure and file counts looked like this:

data/ (2 files)
  images/ (0 files)
    Tree/ (613 files)
    Warrior2/ (527 files)
    DownDog/ (623 files)
    Plank/ (404 files)
  splits/ (0 files)
    train/ (0 files)
      DownDog/ (1000 files)
      Tree/ (1000 files)
      Warrior2/ (1000 files)
      Plank/ (1000 files)
    validate/ (0 files)
      Tree/ (100 files)
      DownDog/ (100 files)

Warrior2/ (100 files)
Plank/ (100 files)
test/ (0 files)
DownDog/ (100 files)
Plank/ (100 files)
Tree/ (100 files)
Warrior2/ (100 files)

Modeling

As is common for image classification, a convolutional neural network was selected as the model type. A base model was built with arbitrarily selected parameters to get a baseline for other models. This model had an accuracy of 89% on the validation data and a relatively low validation loss of 0.3218. Additional models were built through hyper-parameter tuning and the model with the best validation accuracy was fully trained. This model initially had an accuracy of 87% on the validation data set. It was then trained for an additional 100 epochs and the final accuracy was 90%. However, the validation loss on this model was 0.7004. Because of the high loss and chance that it would not generalize well, this model was discarded and the initial base CNN model was selected to move forward.