AsanaVision

Morgan Lo Joseph Ricciardulli

BROWN

CSCI 1430

Motivation

- Since 2020, there has been an unprecedented increase in virtual yoga. These sessions are largely unsupervised making form correction and maintenance challenging.
- Automated yoga pose classification can help to solve this problem by identifying and classifying yoga poses from videos or photos.
- In addition, a yoga pose classifier can provide real time feedback, notifying the user if they are in the correct pose.

The Problem

Classification
This is primarily a classification problem, a satisfactory classifier should be able to identify multiple yoga poses in real time. The difficulty of the problem comes from the unique nature of human bodies and clothing, and their different shapes, sizes, and colors.

Generalization
While it can be relatively straightforward to build a classifier that performs well on identifying yoga poses from a dataset, the dataset may not always capture the intricacies of general use. This complicates model construction and hyperparameter selection.

Goals

1. Custom Yoga CNN Classifier: Build a custom CNN that can accurately classify 5 classes of yoga poses.
2. Model Comparison: Explore the limitations of our custom architecture by measuring the results of three industry standard classification models. The models we will be comparing are EfficientNet-V2, ResNet-18, and YOLOv11. We will also explore data limitations by comparing YOLOv11 performance on a dataset with over 100 classes.
3. Real-Time Demo: Deploy a video pipeline with confidence thresholding and on-screen pose classification for general use.

The Models

State of the Art CNNs
Each of the following models was trained using transfer learning on pre-trained imagenet weights.

Custom CNN
A residual-style CNN that incorporates convolutional blocks, skip connections, and batch normalization.

EfficientNetV2B0 [1]
A lightweight CNN praised for fast training and high accuracy. From tf.keras.applications

ResNet-18 [2]
A CNN based deep neural network using “skip connections” to avoid vanishing gradients. From torchvision.models

YOLOv11 [3]
Latest adaptation of “You Only Look Once.” Object detection and classification single CNN pass. From ultralytics

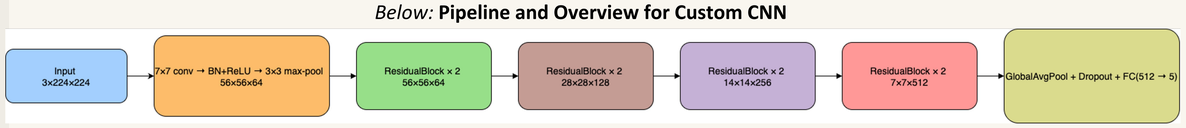
Model	Custom CNN	EfficientNetV2	ResNet-18	YOLOv11n
Parameters	11.1M	7.2M	11.5M	1.6M

Data and Custom CNN Pipeline

Kaggle Yoga Poses Dataset
Over 1.5k images belonging to 5 common yoga pose classes. Includes images of humans and cartoons/drawings. All models were trained on this dataset as a baseline. The relatively small size benefits transfer learning models over the custom CNN.

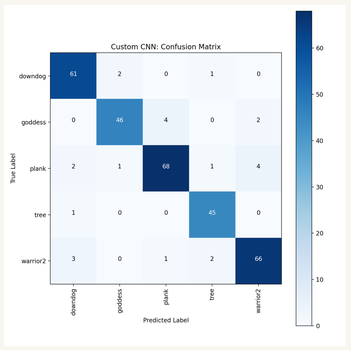
yoga_augmented
A dataset of over 5.5k images belonging to 107 classes. To add robustness, dataset size was tripled using Roboflow, with horizontal flips, blur, and noise. Includes images of humans and cartoons/drawings. YOLO trained on this.

Below: **Pipeline and Overview for Custom CNN**

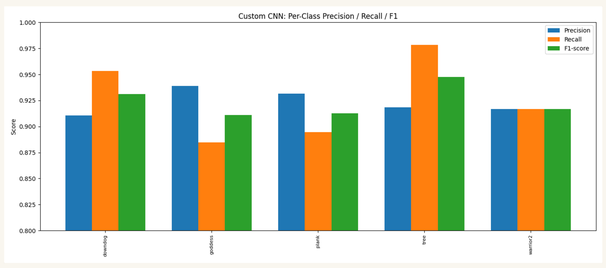


Custom CNN Results

The custom CNN reached a high accuracy, the confusion matrix reflecting this with few incorrect predictions

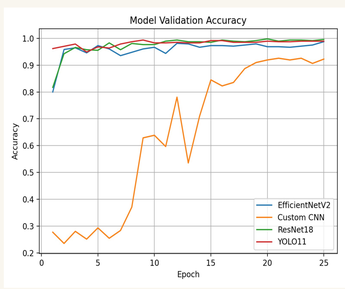


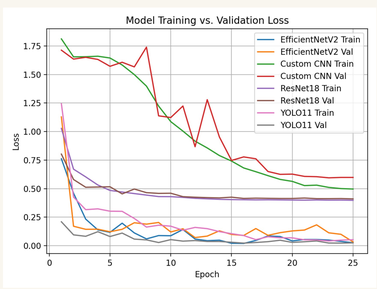
Precision and recall varied per class, tree pose being the most accurate class. Low recall but high precision for goddess and plank indicate a propensity not to pick these classes, while high recall and relatively low precision for downdog and tree indicate a clear preference.



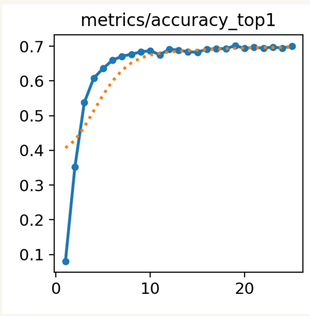
Comparative Results

Transfer learning outperformed the custom CNN – by a lot. Despite fewer parameters (except ResNet), the help of pretrained imagenet weights made a huge difference in accuracy and loss.





Accuracy and loss for state of the art models was similar across the board, EfficientNet and YOLO performing best. Generalizability for Custom CNN and ResNet in DEMO



Right: **YOLO11 transfer Learning on 107 classes**

Applying transfer learning on YOLO11 with the 107 class dataset yielded much worse results, with the highest validation accuracy barely approaching 70%

References

[1] Tan, Mingxing, and Quoc V. Le. **“EfficientNetV2: Smaller Models and Faster Training.”** https://arxiv.org/abs/2104.00298
[2] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. **“Deep Residual Learning for Image Recognition.”** https://doi.org/10.1109/CVPR.2016.90
[3] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. **"You Only Look Once: Unified, Real-Time Object Detection."** https://arxiv.org/abs/1506.02640

Acknowledgements

We would like to thank Professor Tompkin and Kamyar Mirfakhraie!