App Development Journal 2

Aditya Singh, BUSN-208

Overall Goal

The original intent was to create a single iOS app that uses two Apple CoreML packages. That said, I wanted to take things a couple steps further, so I made two apps: the first one uses MNIST (single digit recognition) and YOLOv3 (most prominent object recognition), while the second one involves more UI elements and animations, but only uses MobileNetV2 (object recognition).

App #1: 1_MNIST_YOLOV3

Implementation

Using LLMs, I positioned 4 drawing canvases in a square pattern and loaded each with MNIST capabilities. Then, I instructed the app to take the most probable output from each canvas and multiplied them together to generate a final output.

CoreML Model: MNIST (Single Digit Recognition)

Purpose: Convert four independent single-digit inputs to integer values. Then, multiply the two-digit number formed by the upper two digits (e.g. 7+3=73) by the two-digit number formed by the lower two digits into a single integer value.

Input: Drawings

Output: Integers

Notes: The accuracy of individual digits ultimately depends on how close the representation of the digit is to the baseline established in the MNIST model's training data. This can lead to some inaccuracies in testing.

Implementation

I created a very simple image to text converter that takes in an image and outputs a list of the images that it detects. This has proven to be fairly inaccurate and would need further development at a much more advanced level in order to be completely usable in any real-world context, but the core functionality of YOLO is preserved.

CoreML Model: YOLOv3 (Object Detection)

Purpose: Enter an image and view an itemized list of what it contains.

Input: Image

Output: List of contents

Notes: This has been notoriously buggy on the default pictures of plants that are loaded into the native iOS simulators, so there may be issues in testing.

App #2: 2_MOBILENETV2

Implementation

I used more UI elements (splash pages, animations, etc.) to supplement an implementation of the core functionality of MobileNetV2 (object detection). Once the user follows the given instructions and inputs an object, the app will output an emoji that most closely corresponds to the object and will display a description of what it saw, a slider that rœepresents its confidence level, and a figure that represents the confidence level (e.g. 98%).

CoreML Model: MobileNetV2 (Object Recognition)

Purpose: Enter an image and receive a single output of what the object contains, as well as an emoji (from a limited set) and a level of confidence.

Input: Image

Output: Single item

Notes/Accuracy: This has been…somewhat accurate throughout my testing. It is very consistent from input to input, but continues to fail to recognize plants.