Saeed Arabi, email: arabi@iastate.edu. phone:5157085857

Cell phone detection app using hybrid of deep-learning model and classical computer vision method.

Build instructions:

- `train_phone_finder.py` will download the deep learning object detection model (faster rcnn nas) trained on COCO dataset and store it in `~/.keras/datasets`. This is the only file needed for inference.

General notes about the project:

- The application was tested on these two machines and the consistent accuracy of the model were confirmed:
- MAC OS BIG SUR with tensorflow-cpu 2.4.1 and opencv 4.5.1.
- Ubuntu 18.04 with tensorflow-gpu 2.4.1 and opencv 3.1.0.
- The object detector class implemented in the `helper.py`.
- The methods and attributes names were chosen to be easy to follow and to minimize the requirement for comments.
- The main detection process can be found in the `feed` method of the detector class. First, deep learning object detection method used to detect the phone. If it the detection was not successful, the classical computer vision solution, using color filtering, was used.
- objects detected by color filtering were filtered by the area of the bounding box.
- The accuracy of the algorithm is **88.37%**. The `generate_detection_results.py` script will draw bounding box center on the image and out put the detailed results. You may find its output in the **Results** section.

Future work:

- I just developed a workable prototype for this task. This model is not optimized for speed nor accuracy by any means. Considering the simple nature of the dataset, we should be able get more data, annotate them and train our custom model on that to reduce the false positives/negatives. For this approach, I would use a light detector such as SSD-mobilenet which is suitable for small-size datasets.

- The other approach is to use sliding windows with different sizes and scales and use ***HOG*** and color histograms to get the features of each cropped bounding boxes and feed the results to a ***SVM*** classifier. Using shallow models like this is specifically useful for this application as the shape and color of the cell phone is highly distinct from non-cell phone objects in the dataset.

Results:

correct counts: 114

incorrect counts: 15

accuracy: 88.37 %

files_detected_by_deep_model: [‘./find_phone/128.jpg’, ‘./find_phone/3.jpg’, ‘./find_phone/106.jpg’, ‘./find_phone/102.jpg’, ‘./find_phone/36.jpg’, ‘./find_phone/6.jpg’, ‘./find_phone/52.jpg’, ‘./find_phone/11.jpg’, ‘./find_phone/74.jpg’, ‘./find_phone/31.jpg’, ‘./find_phone/12.jpg’, ‘./find_phone/111.jpg’, ‘./find_phone/110.jpg’, ‘./find_phone/44.jpg’, ‘./find_phone/73.jpg’, ‘./find_phone/132.jpg’, ‘./find_phone/69.jpg’, ‘./find_phone/94.jpg’, ‘./find_phone/113.jpg’, ‘./find_phone/107.jpg’, ‘./find_phone/101.jpg’, ‘./find_phone/76.jpg’, ‘./find_phone/71.jpg’, ‘./find_phone/67.jpg’, ‘./find_phone/57.jpg’, ‘./find_phone/20.jpg’, ‘./find_phone/25.jpg’, ‘./find_phone/79.jpg’, ‘./find_phone/68.jpg’, ‘./find_phone/0.jpg’, ‘./find_phone/130.jpg’, ‘./find_phone/22.jpg’, ‘./find_phone/92.jpg’, ‘./find_phone/117.jpg’, ‘./find_phone/98.jpg’, ‘./find_phone/122.jpg’, ‘./find_phone/126.jpg’, ‘./find_phone/50.jpg’, ‘./find_phone/84.jpg’, ‘./find_phone/129.jpg’, ‘./find_phone/62.jpg’, ‘./find_phone/15.jpg’, ‘./find_phone/23.jpg’, ‘./find_phone/85.jpg’, ‘./find_phone/87.jpg’, ‘./find_phone/99.jpg’, ‘./find_phone/37.jpg’, ‘./find_phone/86.jpg’, ‘./find_phone/105.jpg’, ‘./find_phone/26.jpg’, ‘./find_phone/7.jpg’, ‘./find_phone/78.jpg’, ‘./find_phone/35.jpg’, ‘./find_phone/119.jpg’, ‘./find_phone/127.jpg’, ‘./find_phone/88.jpg’, ‘./find_phone/96.jpg’, ‘./find_phone/121.jpg’, ‘./find_phone/123.jpg’, ‘./find_phone/125.jpg’]

files_detected_by_color_filtering: [‘./find_phone/24.jpg’, ‘./find_phone/5.jpg’, ‘./find_phone/17.jpg’, ‘./find_phone/4.jpg’, ‘./find_phone/39.jpg’, ‘./find_phone/46.jpg’, ‘./find_phone/29.jpg’, ‘./find_phone/49.jpg’, ‘./find_phone/61.jpg’, ‘./find_phone/91.jpg’, ‘./find_phone/51.jpg’, ‘./find_phone/63.jpg’, ‘./find_phone/53.jpg’, ‘./find_phone/131.jpg’, ‘./find_phone/109.jpg’, ‘./find_phone/9.jpg’, ‘./find_phone/16.jpg’, ‘./find_phone/30.jpg’,

‘./find_phone/33.jpg’, ‘./find_phone/134.jpg’, ‘./find_phone/70.jpg’,
‘./find_phone/38.jpg’, ‘./find_phone/32.jpg’, ‘./find_phone/120.jpg’,
‘./find_phone/80.jpg’, ‘./find_phone/124.jpg’, ‘./find_phone/27.jpg’,
‘./find_phone/83.jpg’, ‘./find_phone/104.jpg’, ‘./find_phone/72.jpg’,
‘./find_phone/13.jpg’, ‘./find_phone/77.jpg’, ‘./find_phone/115.jpg’,
‘./find_phone/14.jpg’, ‘./find_phone/55.jpg’, ‘./find_phone/114.jpg’,
‘./find_phone/64.jpg’, ‘./find_phone/90.jpg’, ‘./find_phone/45.jpg’, ‘./find_phone/100.jpg’,
‘./find_phone/42.jpg’, ‘./find_phone/93.jpg’, ‘./find_phone/95.jpg’, ‘./find_phone/81.jpg’,
‘./find_phone/54.jpg’, ‘./find_phone/112.jpg’, ‘./find_phone/89.jpg’,
‘./find_phone/108.jpg’, ‘./find_phone/10.jpg’, ‘./find_phone/116.jpg’,
‘./find_phone/66.jpg’, ‘./find_phone/82.jpg’, ‘./find_phone/8.jpg’, ‘./find_phone/133.jpg’]

files_with_no_detection: [‘./find_phone/1.jpg’, ‘./find_phone/59.jpg’,
‘./find_phone/40.jpg’, ‘./find_phone/47.jpg’, ‘./find_phone/18.jpg’, ‘./find_phone/48.jpg’,
‘./find_phone/118.jpg’, ‘./find_phone/58.jpg’, ‘./find_phone/75.jpg’,
‘./find_phone/34.jpg’, ‘./find_phone/43.jpg’, ‘./find_phone/60.jpg’, ‘./find_phone/41.jpg’,
‘./find_phone/103.jpg’, ‘./find_phone/97.jpg’]