

Discussion

My approach to this problem was to use blob detection with OpenCV library, which includes the following steps:

- Blurring and thresholding input image
- Finding contours in the thresholded image and drawing a smallest enclosing circle around each of them. The center of the enclosing circle defines the location of the corresponding contour/object.
- Calculating a score for each contour based on its area and perimeter and returning the location of the contour with the highest score.

Using the approach described above, phone location can be detected in 96+% of the images in the provided dataset (3.88% error rate). The only optimization I've utilized was looping through the possible space of target_area and target_perimeter values to find values most likely to contain a cell phone.

Considering the small sample size and undemanding (70%) detection requirements, machine learning with CNNs would likely not be the most efficient way to meet the required detection accuracy. That said, ML and simpler methods such as contour or edge detection are not mutually exclusive. One way to combine them would be to use these simpler methods for image preprocessing and train a CNN on the modified images. Another way would be to use openCV to extract various features from the image (contours, edges, etc) and use those features as an input to a ML classifier.