one
tiny
model

the outline

the proposal

(or, background and data)

the goal

Retrieve each photo from one tiny hand and use image processing techniques and machine learning to detect the one tiny hand.

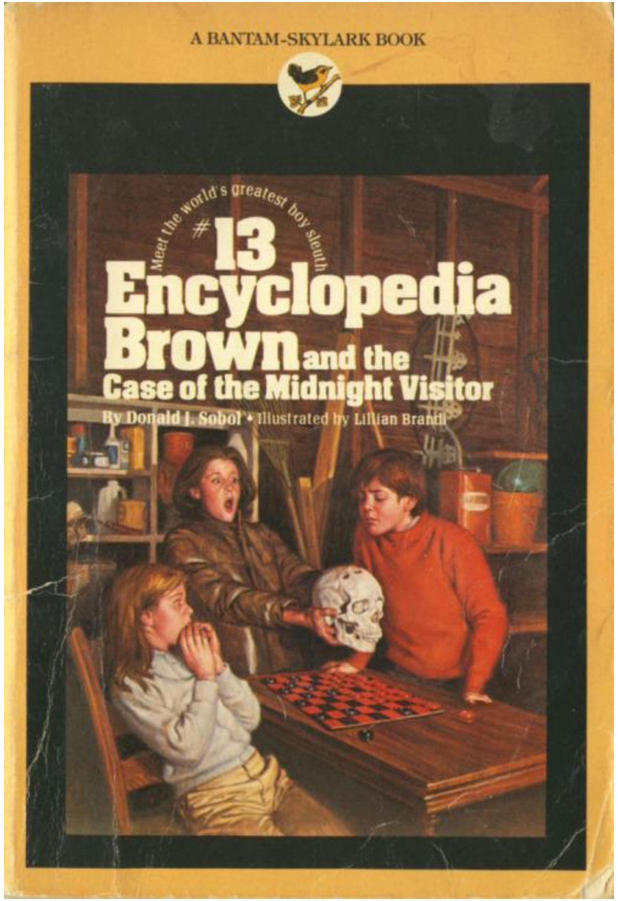
the objectives

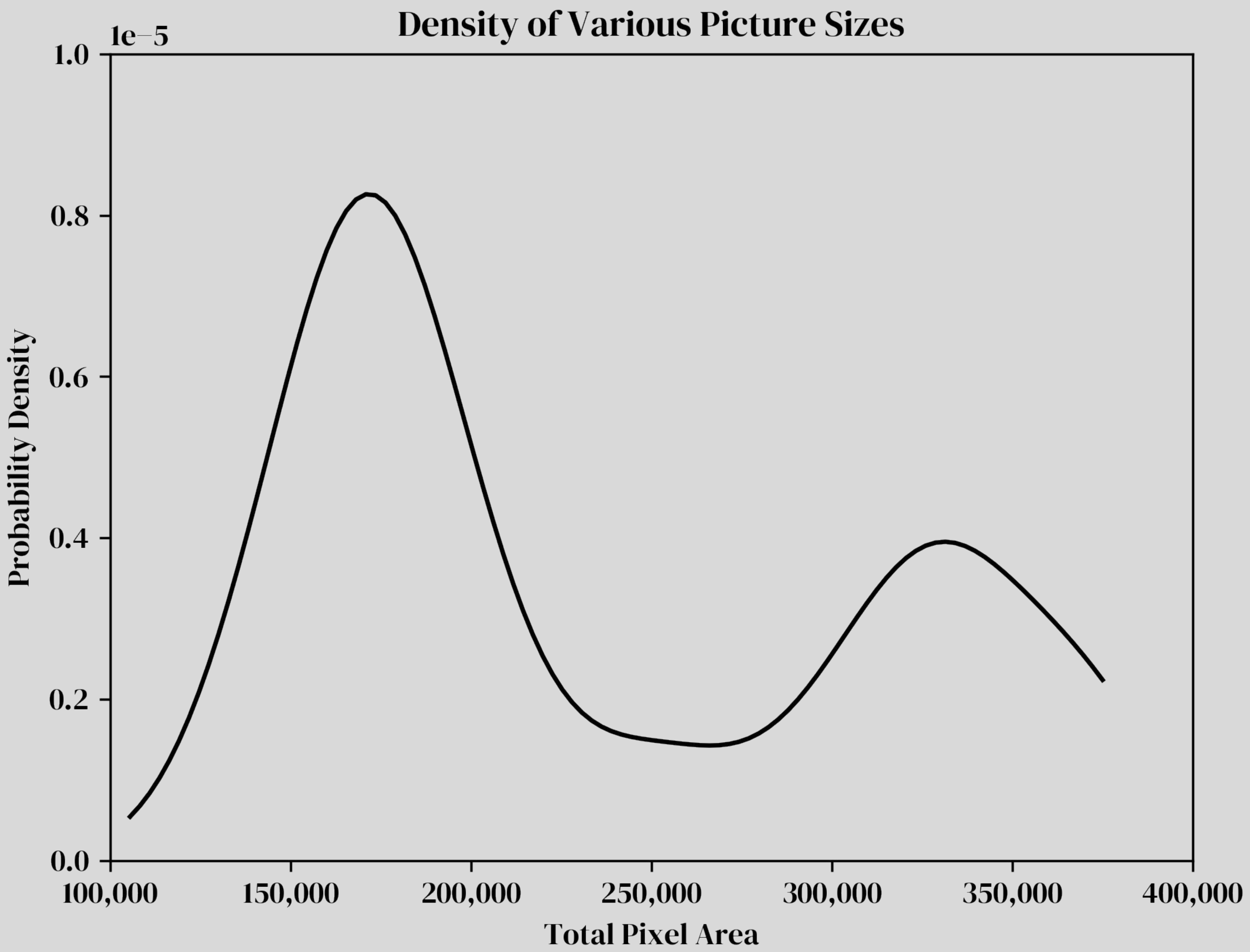
To learn image processing and object detection machine learning techniques, various modules in Python like scikit-image, OpenCV, tensorflow and keras, and object-oriented programming skills.

the data

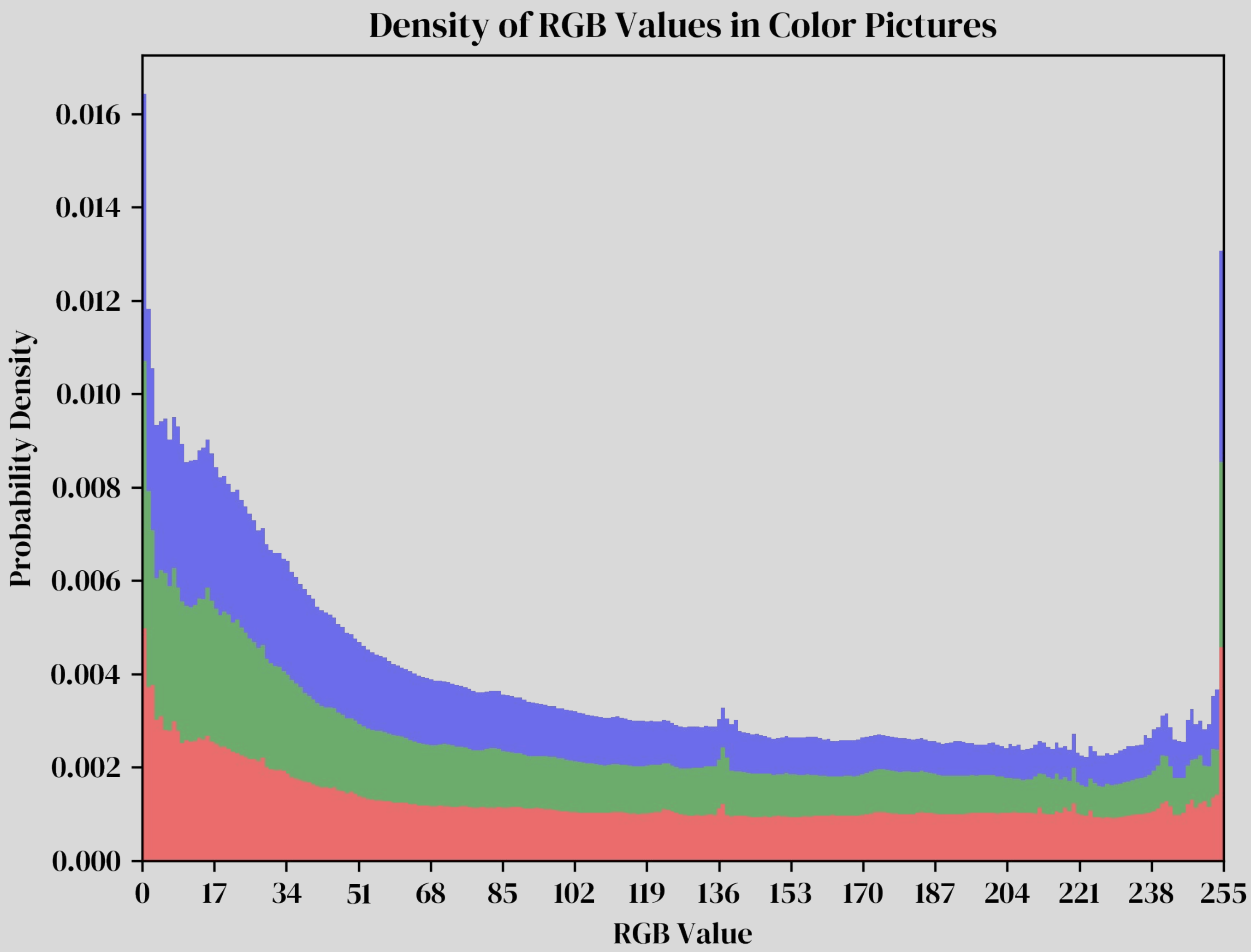
- The data consisted of ~675 pictures from one tiny hand.
- ~100 of these were excluded. Examples follow on the next slide.
- The vast majority (~568) of the remainder were in color: only a few were in black and white.
- The vast majority of pictures were 500 pixels wide. The variation in size among them is almost exclusively a function of their height.

the exclusions





Density of Various Picture Sizes
1e−5
1.0
0.8
0.6
0.4
0.2
0.0
Probability Density
100,000
150,000
200,000
250,000
300,000
350,000
400,000
Total Pixel Area

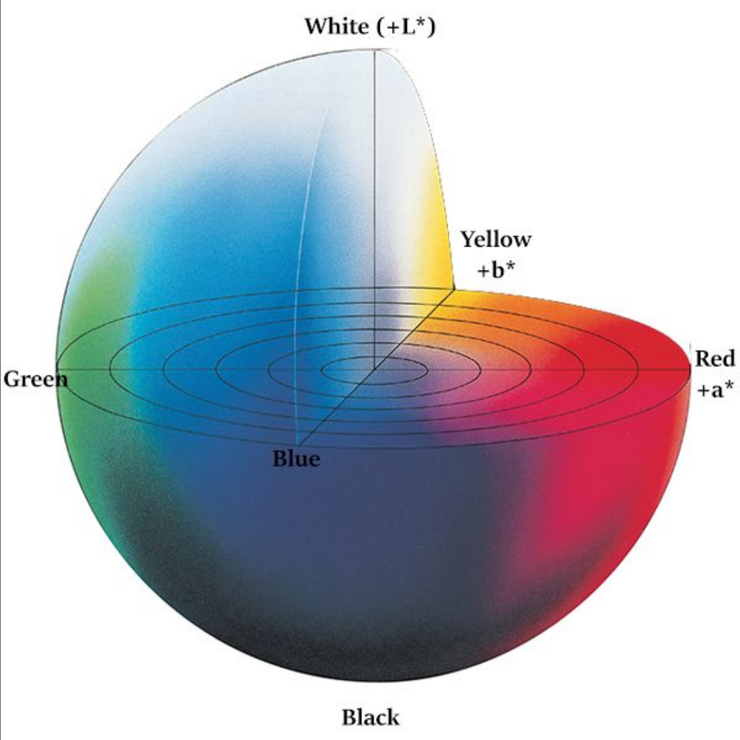

Density of RGB Values in Color Pictures
Probability Density
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
17
34
51
68
85
102
119
136
153
170
187
204
221
238
255
RGB Value

the journey

(or, unsupervised learning)

the big idea

Use image processing techniques in scikit-image to segment the one tiny hand from the other features of an image.

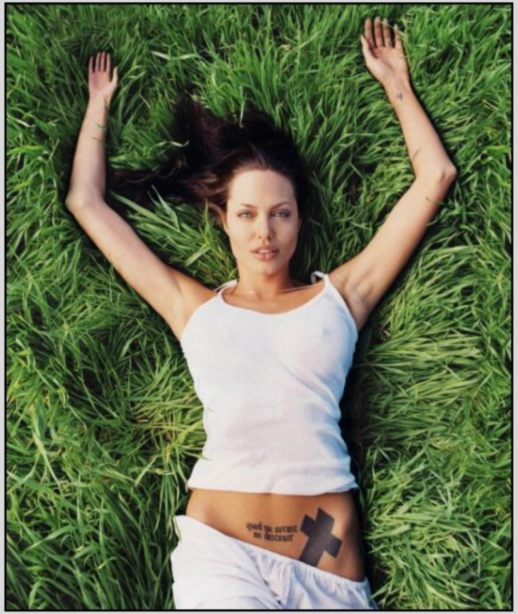


slic

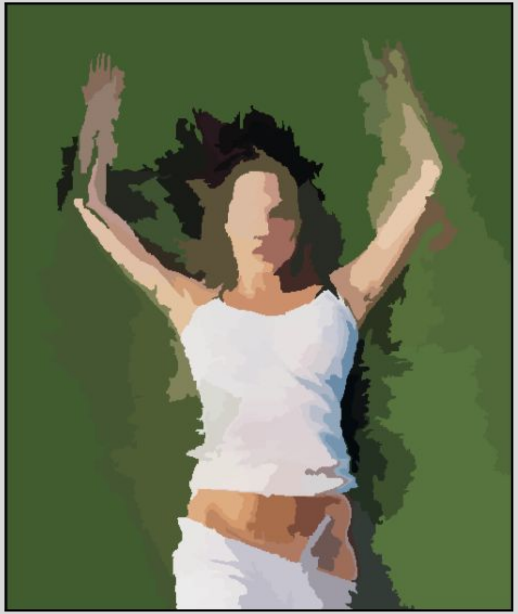
- Simple Linear Iterative Clustering (SLIC) creates superpixels–or rather, clusters–based on color intensity and proximity to generate a lower dimensional representation of an image.

- The goal is to segment images but reduce instances where two segments bleed into each other and improve boundary recall–where edges of superpixels match actual picture edges.

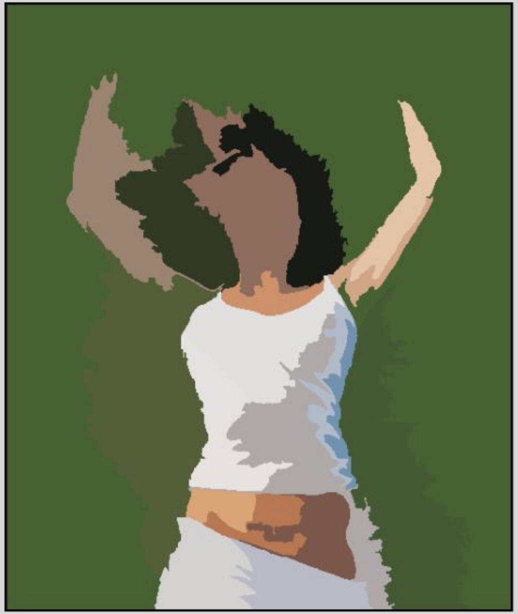
Original



SLIC Segmentation - 500 Superpixels



SLIC Segmentation - 100 Superpixels

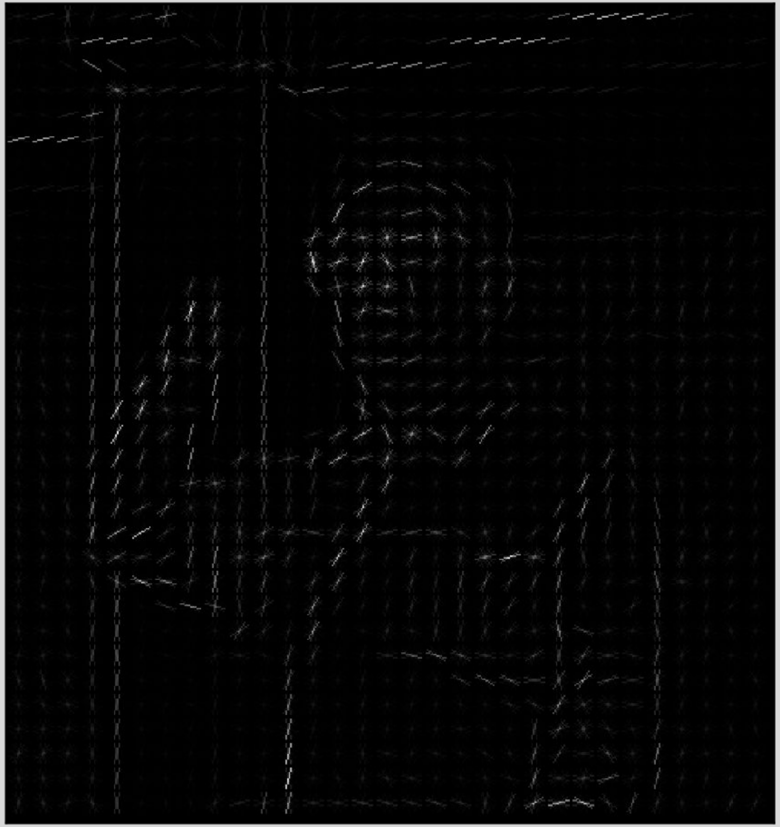


- Histograms of Oriented Gradients (HOGs) describe an image by binning image pixels in accordance with the direction of their gradients, or the magnitude of changes in pixel intensity.
- HOGs can also be contrast-normalized to reduce the effects of lighting or shadows in an image.

hog

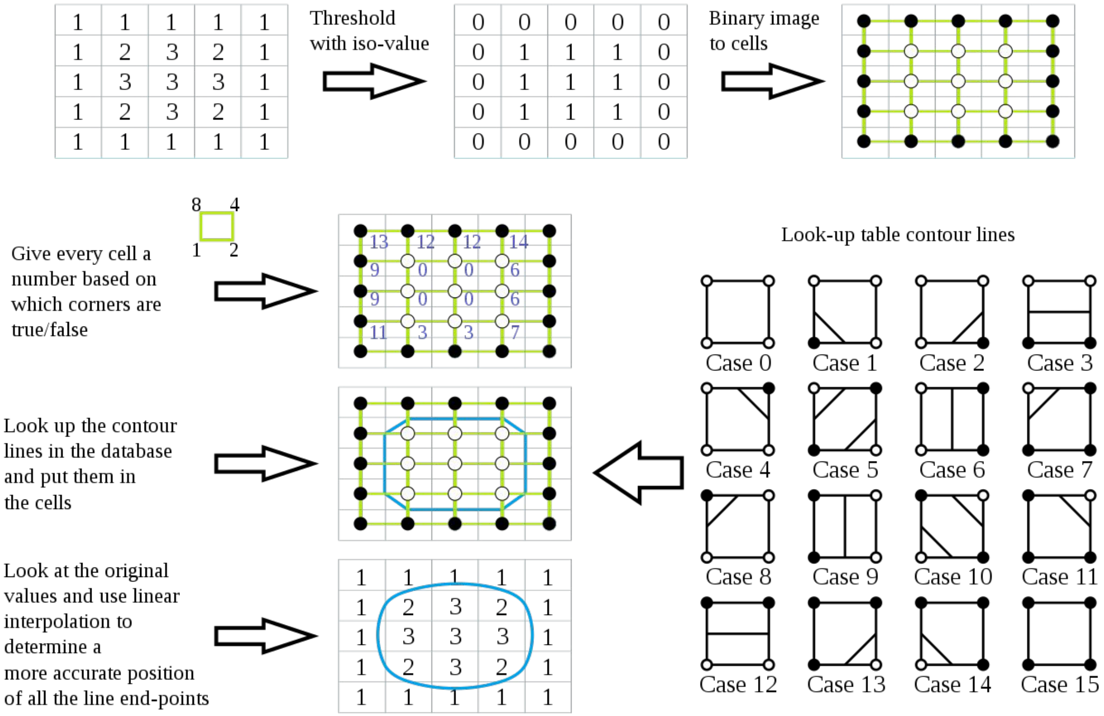
Original

HOG Version



contours

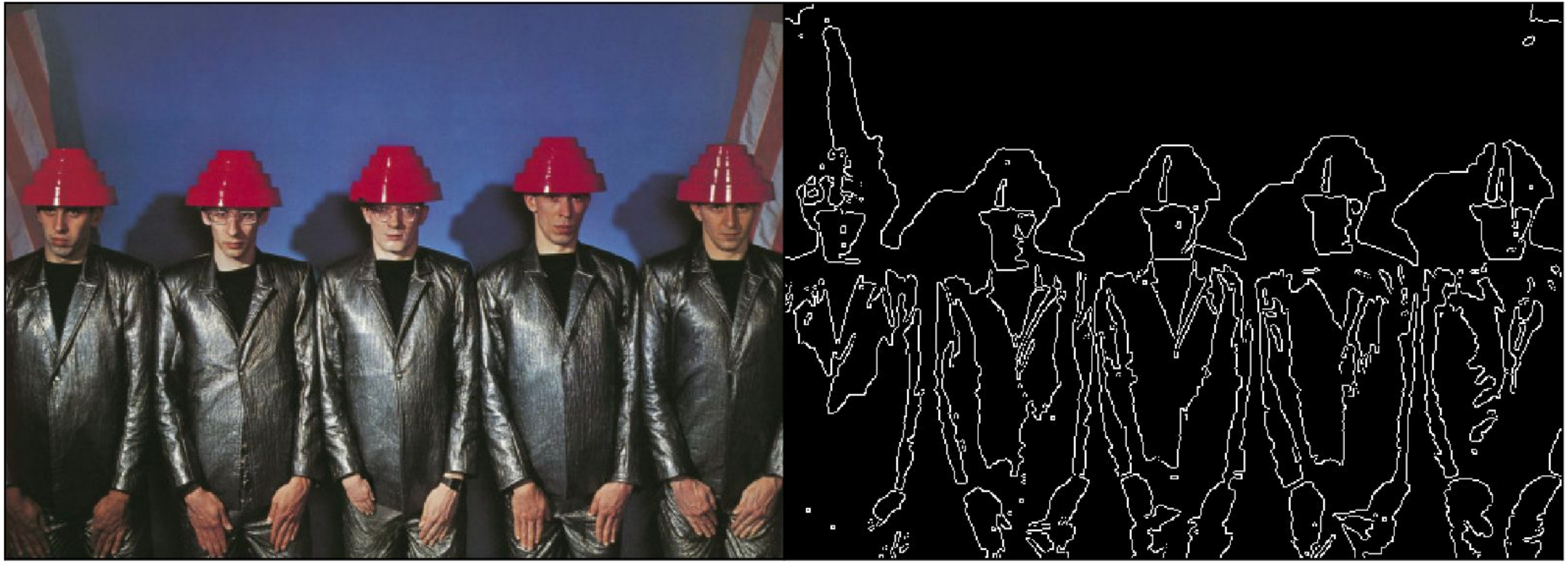
There are many methods for edge or contour detection: one of those used by scikit-image is the marching squares algorithm, as described in this figure.



Original

With Contours

Original **Edges Detected by Canny**



edges

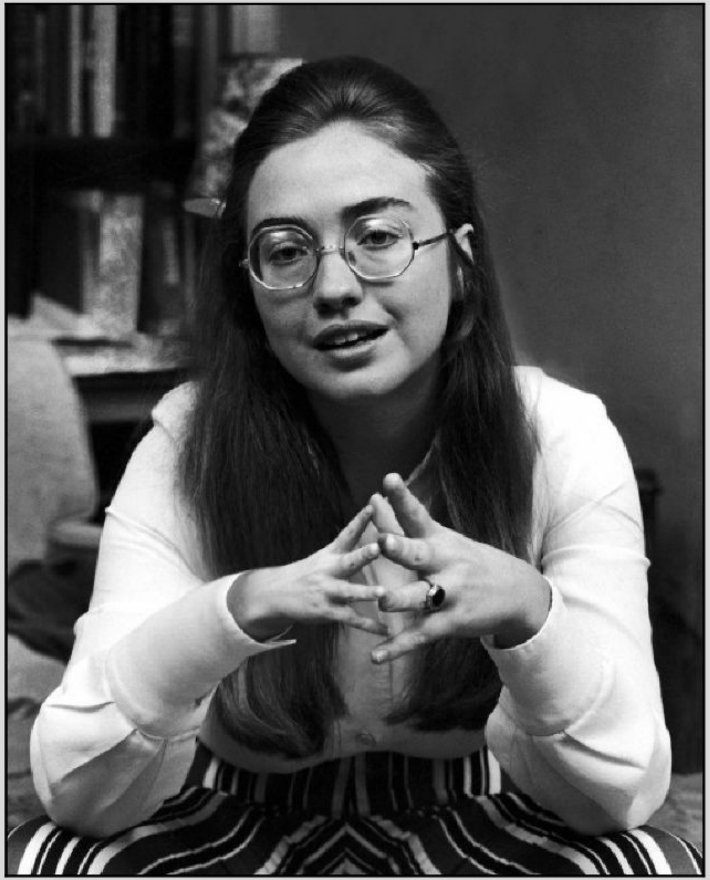
Scikit-Image and OpenCV also have a Canny edge detector, which applies a Gaussian filter to blur the image and then further filters out “noise” edges by a series of steps.

- Image blobs (or binary large objects) are groups of pixels that share a common property that is different than the surrounding pixels.

- There are many methods to finding blobs in an image, but the following slide showcases the Determinant of the Hessian (DOH).

blobs

Original

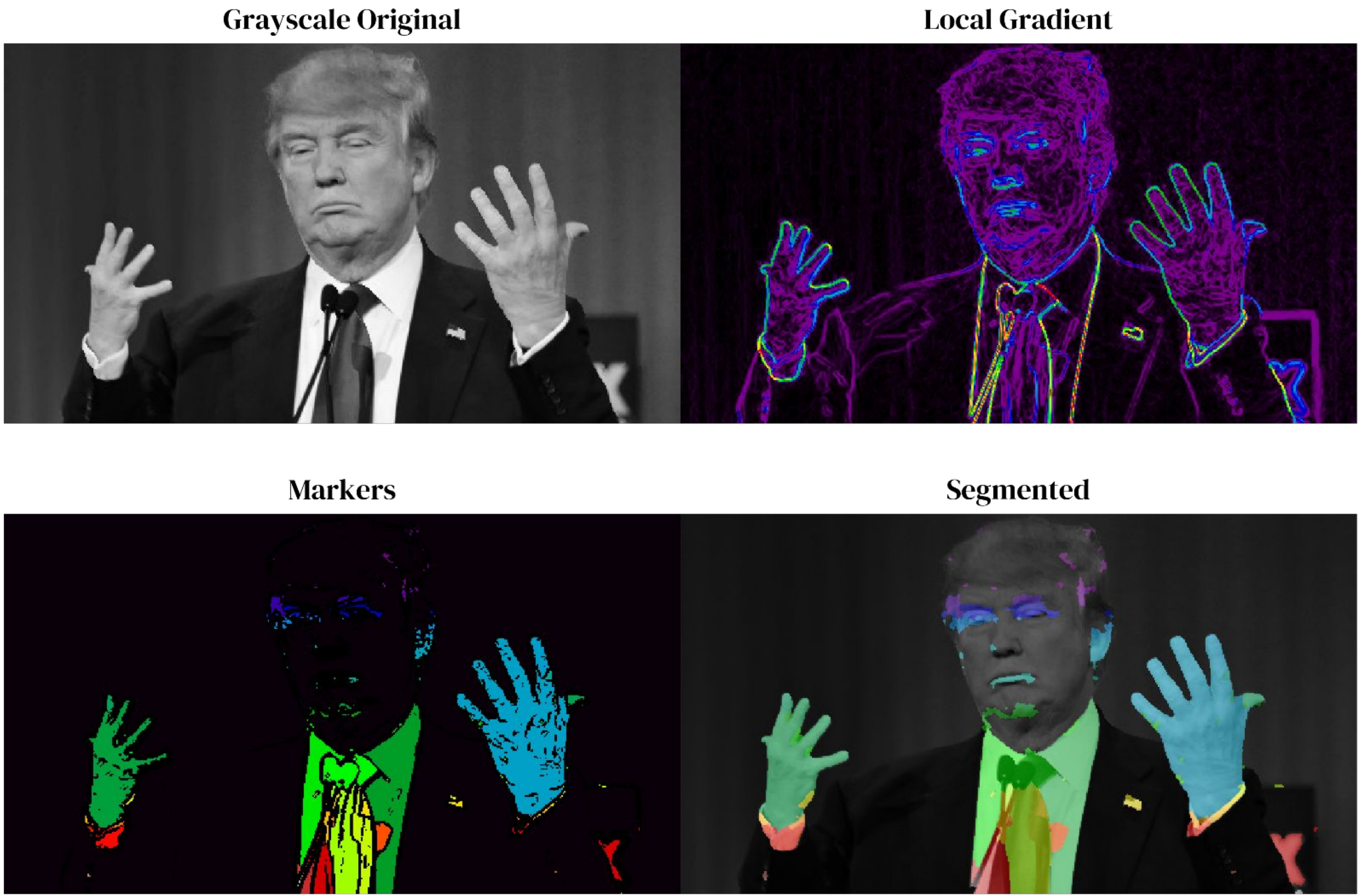


DOH Blobs

watershed

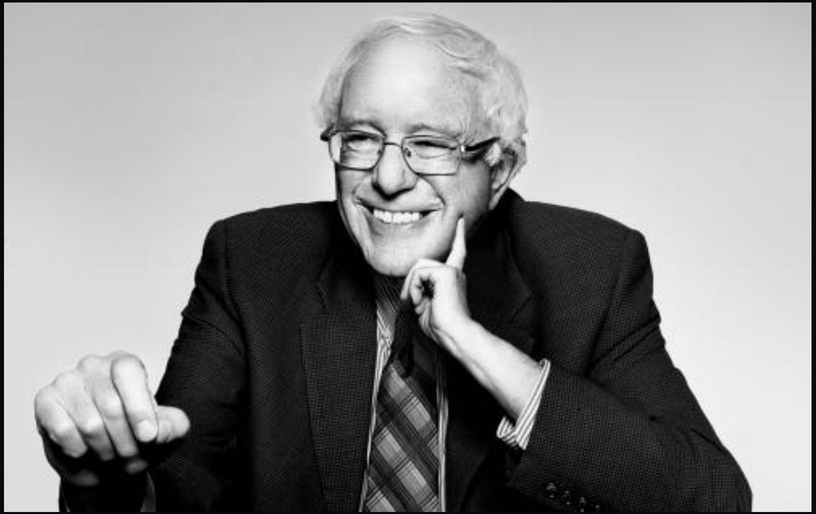
- Another method of segmentation is called watershed. Images are treated as topographies, where pixel intensities are “peaks” or “valleys” and the “valleys” are flooded.

- The flooding results in segmentation between foreground and background.


Grayscale Original
Local Gradient
Markers
Segmented

the models

(or, supervised learning)



another idea

Use machine learning models to predict not only bounding boxes for hands but also whether the hand in the bounding box was normal or tiny.

- Images were labelled with bounding boxes using LabelImg using YOLO coordinates. These coordinates were scaled or converted as needed.
- For each model, pictures were imported with their coordinates, resized as optimal for that model type, converted to grayscale and contrast-enhanced.
- Pre-trained models (i.e., trained on COCO) were then trained, validated and tested on these pictures.

the setup

$$IoU(A,B) = \frac{A \cap B}{A \cup B}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$AP = \frac{1}{11} \sum_{r=0}^{10} P_{\frac{r}{10}}$$

map

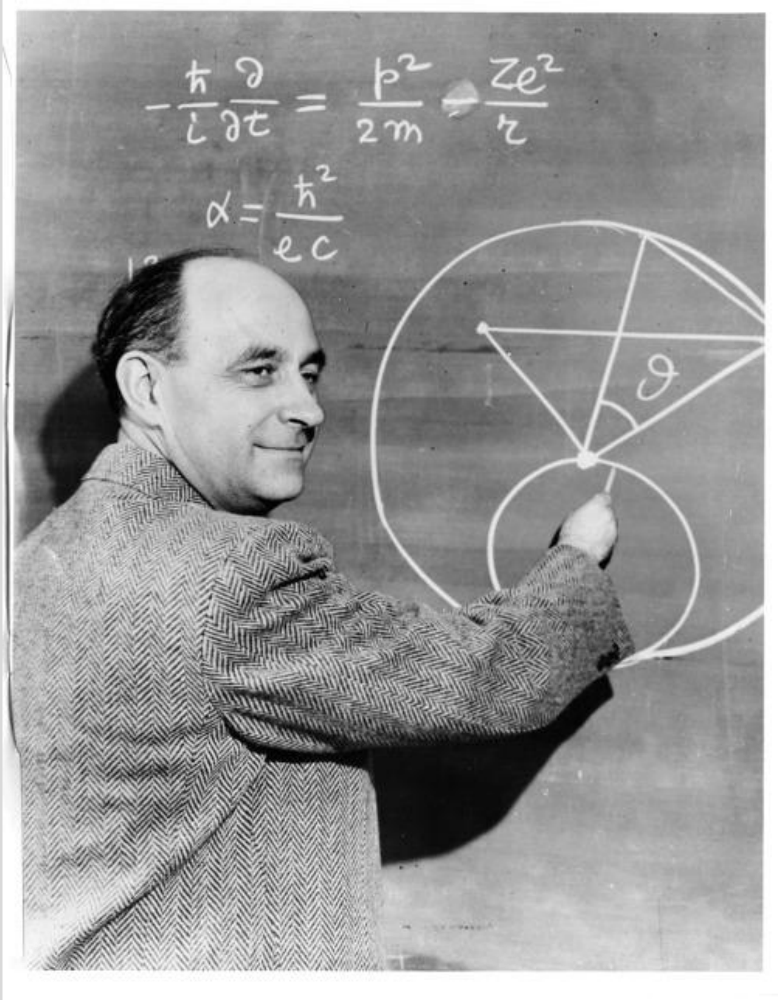
For object detection models, mean average precision (mAP) is the primary quality metric used. An intersection of union threshold (usually, 0.5) is chosen, and the mean of the average precision for each class at each 0.1 increase in recall is calculated.

$$mAP = \frac{1}{n} \sum_{k=1}^{n} AP_k$$

mask rcnn theory

- The first model used was Mask RCNN.

- Mask RCNN steps (high-level):
  - Create feature map.
  - Select regions of objectness based on anchor boxes and refine bounding boxes.
  - Classify regions as particular objects and further refine bounding boxes.
  - Generate masks for bounding box shapes.

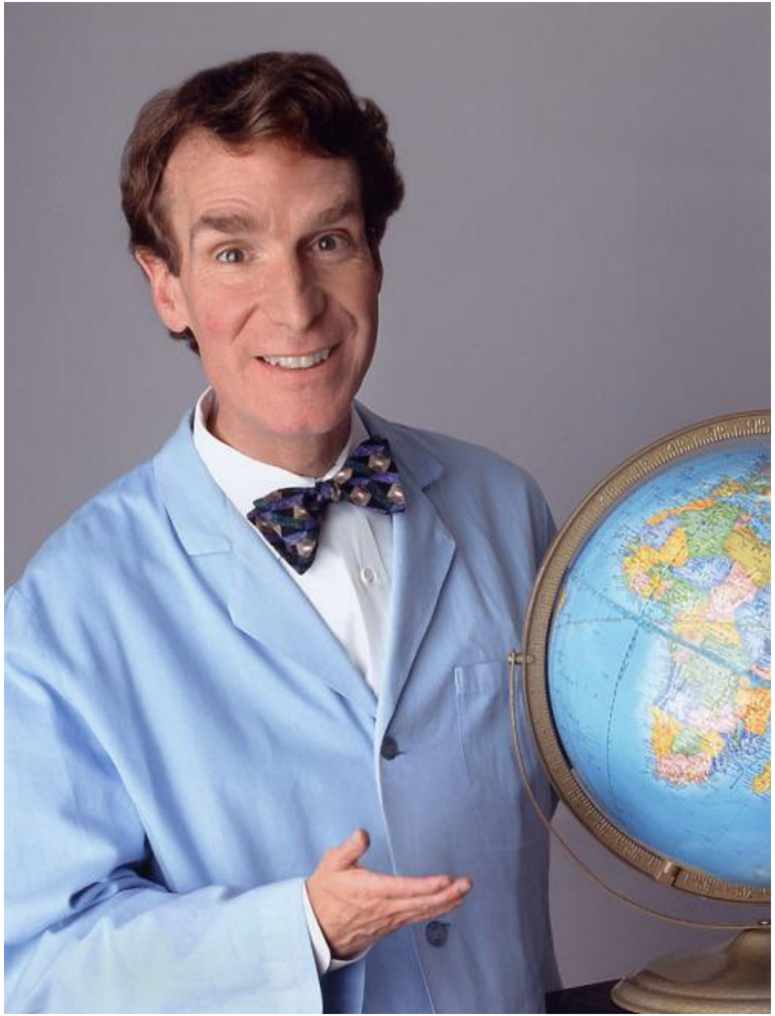


mask rcnn results

- Generally, the Mask RCNN model performed poorly. mAP on test set was quite low: 0.403.

- The model may need more training–e.g., with more samples, or with additional epochs.

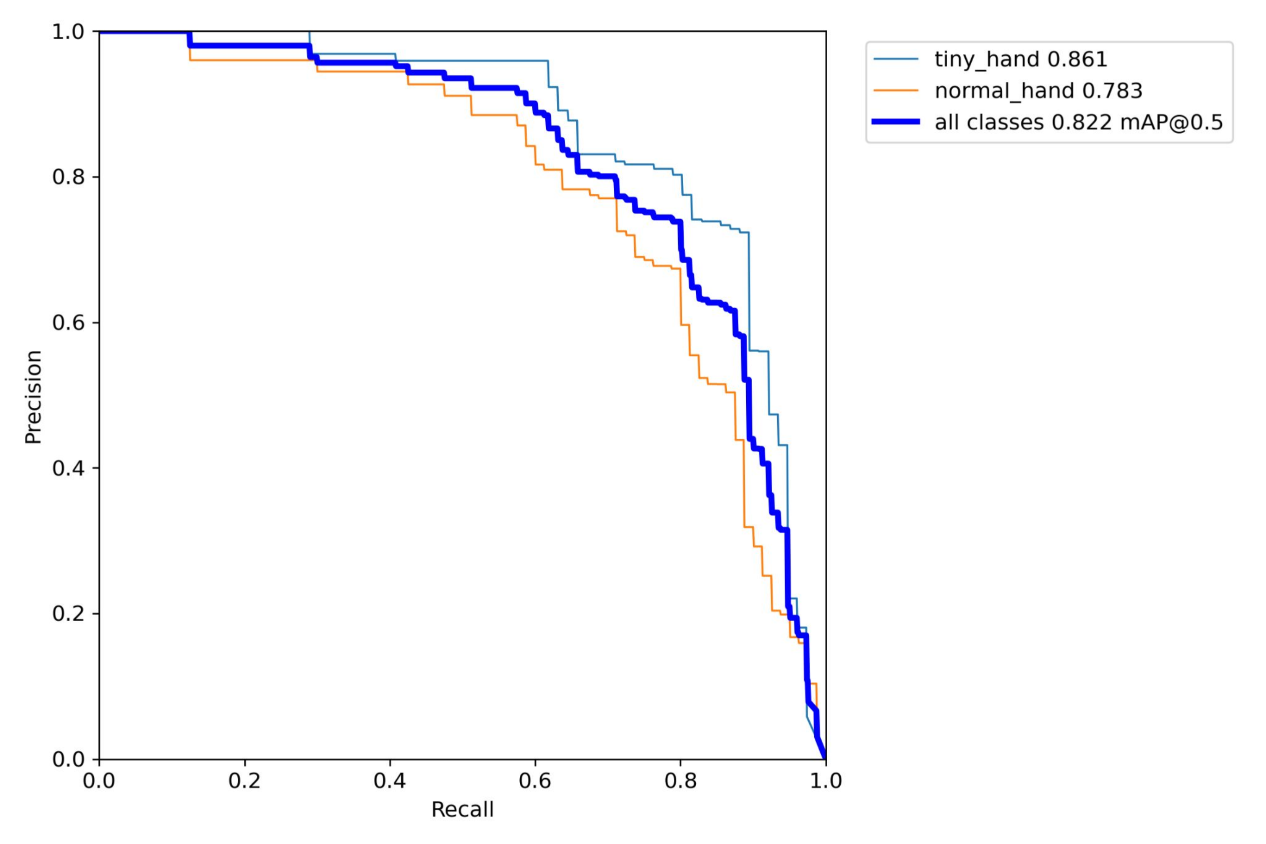
- Alternatively, it may need better anchor boxes.



yolov5 theory

- Conceptually, You Only Look Once (YOLO) models have a similar structure to Mask RCNN models, but bounding boxes and classifications within YOLO models are determined at the same time. This characteristic makes YOLO models much faster and capable of real-time object detection (e.g., in videos).

- As a trade-off for speed, YOLO models may have lower accuracy, particularly on smaller objects.

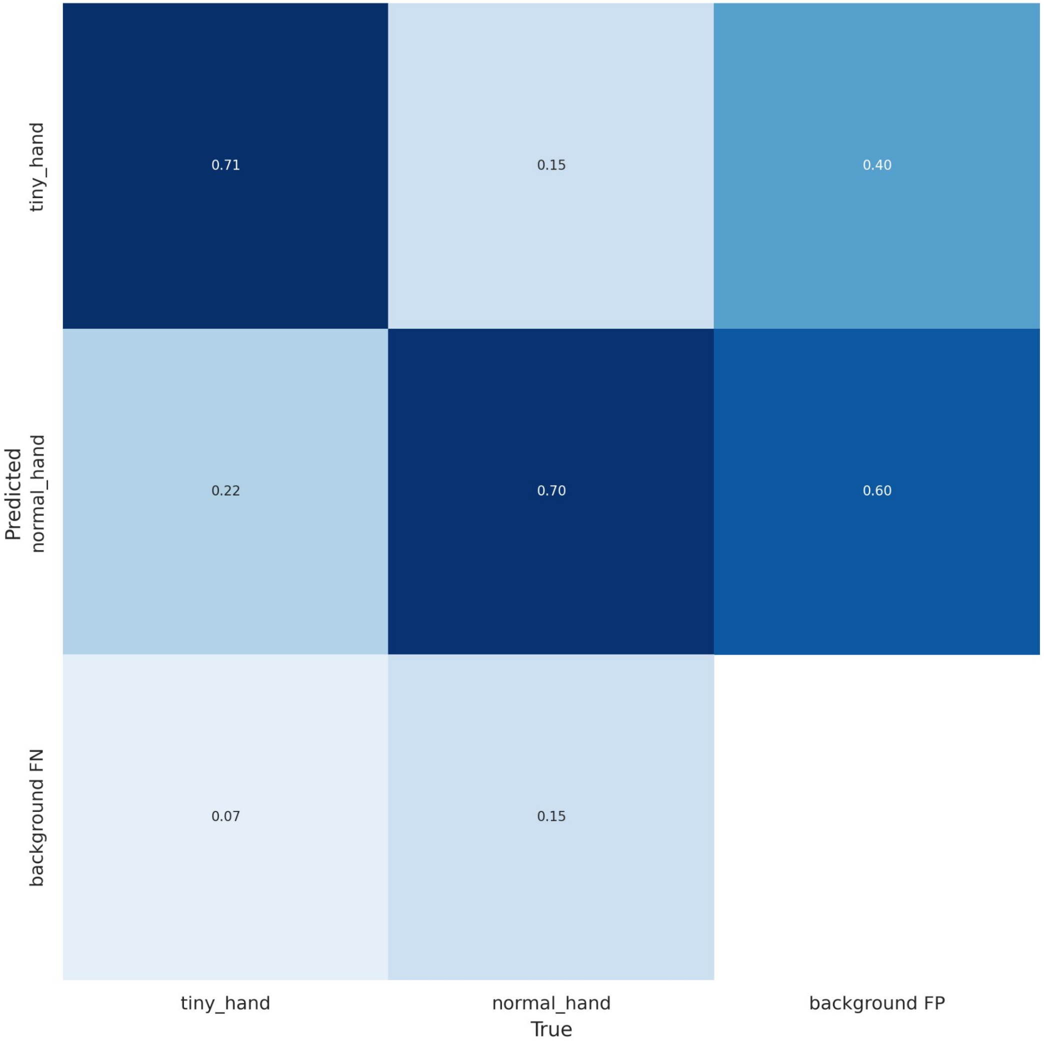


yolov5 results - pr curve

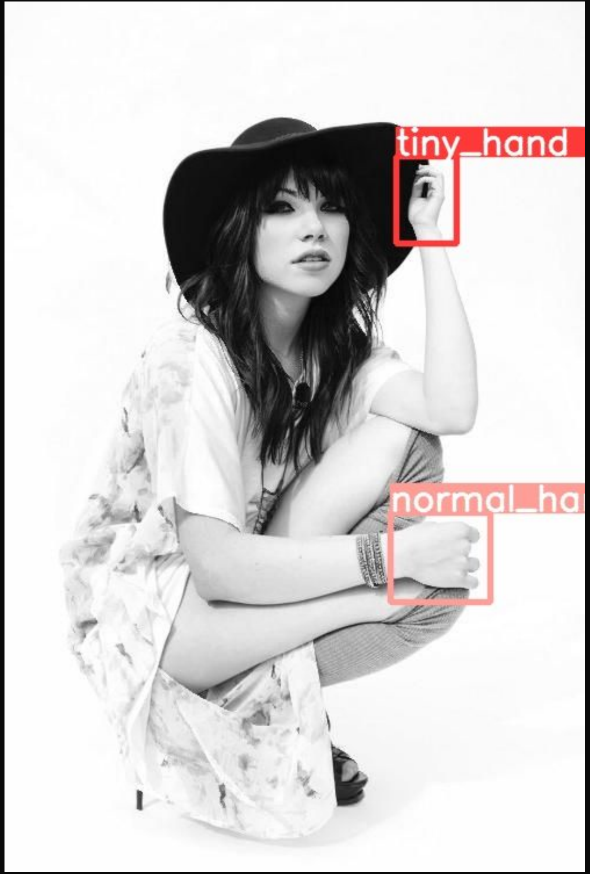
yolov5 results - confusion matrix

- The YOLOv5 model performed better overall than the Mask RCNN model.

- Some struggles:
    - Distinguishing hand classes where multiple hands were in the picture and on top of each other.
    - Accurately detecting hands from background or other objects.



yolov5 results - examples



closing thoughts

- More testing, validation and explanation of the models' performance are needed.

- Further exploration would include:
  - Deeper dives into both one-step and two-step models, including image manipulation detection.
  - Modification of input images, training time and anchor box parameters to achieve better results.
  - Additional segmentation and feature description or extraction techniques.

thank you

Any questions?

Contact Information:
charre2021@outlook.com
www.github.com/charre2021

Special thanks also to:
- Michael Holloway
- Veronica Ikeshoji-Orlati
- Alvin Wendt
- DS5 Cohort
- NSS
- And Joshua Rio-Ross specifically for introducing me to One Tiny Hand