Computer Vision/ Image Processing/ OpenCV(cv2)

(Top branch)
- imread - read an image using this function
- imshow - show an image using this function
- imwrite - write an image to the filesystem using this function
- To access a pixel : (b,g,r) = image[y, x]

images (Note: In opencv an image is represented as a NumPy array)
- shape - provides the shape or properties of an image [rows, columns, channels]
- grayscale image
  - pixel range 0 to 255
  - 0 represents Black
  - 255 represents White
- color image
  - pixel comprises of RGB ([0 to 255], [0 to 255], [0 to 255])
  - (0, 0, 0) represents Black
  - (255, 255, 255) represents White

- OpenCV stores the images in BGR order

- draw a geometric shape on the image
  - cv2.line
  - cv2.rectangle
  - cv2.circle

Image Processing Transformations
- scale images
  - Interpolation
  - Aspect Ratio has to be maintained to avoid skewness and distortion
- crop images
- feature extraction
- bitwise opration
- masking
- Translation or Shifting
  - Row-Major Form
  - Column-Major Form
- Rotation
  - Counter clockwise is +ve angle
  - clockwise is the -ve angle
- Reflections
- Skeletonization
- Flipping
  - 1 to flip on horizontal axis
  - 0 to flip on vertical axis
  - -1 to flip on both horizontal and vertical axes
- Image Matrix Arithmetics
  - using NumPy arrays
  - using CV2
- Splitting & Merging — usefule in thresholding and edge detection
- Kernels — Kernels are used to perform convolutions on the image matrix

Morphological Operations - Morphological operations are commonly used as pre-processing steps to more powerful computer vision solutions such as OCR, Automatic Number Plate Recognition (ANPR), and barcode detection.
- Used to increase or decrease the size of objects in images
- Used to close and open gaps between objects
- Erosion — useful for removing small blobs from an image or disconnected two connected components
- Dilation — Dilations increase the size of foreground object and are especially useful for joining broken parts of an image together.
- Opening — An opening is an erosion followed by a dilation.
- Closing — The exact opposite to an opening would be a closing. A closing is a dilation followed by an erosion.
- Morphological Gradient — morphological gradient is the difference between the dilation and erosion. It is useful for determining the outline of a particular object of an image
- Black Hat
- Top Hat or White Hat

Smoothing
- Blurring
  - averaging — This is finding an average of the matrix data
  - Gaussian blurring — Gaussian blurring is similar to average blurring, but instead of using a simple mean, we are now using a weighted mean, where neighborhood pixels that are closer to the central pixel contribute more "weight" to the average. And as the name suggests, Gaussian smoothing is used to remove noise that approximately follows a Gaussian distribution. Gaussian blur is slightly slower than a simple average blur (and only by a tiny fraction), a Gaussian blur tends to give much nicer results, especially when applied to natural images.
  - median filtering — the median blur method has been most effective when removing salt-and-pepper noise
  - bilateral filtering — In order to reduce noise while still maintaining edges, we can use bilateral blurring. Bilateral blurring accomplishes this by introducing two Gaussian distributions. The first Gaussian function only considers spatial neighbors. That is, pixels that appear close together in the (x, y)-coordinate space of the image. The second Gaussian then models the pixel intensity of the neighborhood, ensuring that only pixels with similar intensity are included in the actual computation of the blur.

Thresholding - It is the most basic Segmentation technique to seperate out the images foreground and background
- Simple Thresholding - Manually supply parameters to segment the image. Works well in controlled lighting conditions where we have high contrast between foreground and background.
- Otsus Thresholding - Dynamic and automatically compute the optimal threshold value based on input image
- Adaptive Thresholding - Instead of trying to apply thersholding globally this technique will apply the thresholding at each pixel level.

Gradients
- These are used to find edges in the image (Edge Detection)
  - The Edges will help us to create contours and outlies of the objects
  - Histogram of Oriented Gradients and SIFT are built upon image gradient representations
  - Gradient images are even used to construct saliency maps, which highlight the subjects of an image.
- Directions
  - North: I(x, y - 1)   South: I(x, y + 1)   East: I(x+1, y)   West: I(x - 1, y)
  - Vertical Change or change in Y-Axis == $G_{y} = I(x, y + 1) - I(x, y - 1)$ Note: I is the image
  - Horizontal Change or change in X-Axis == $G_{x} = I(x + 1, y) - I(x - 1, y)$ Note: I is the image
- Gradient Magnitude — $G = \sqrt{G_{x}^{2} + G_{y}^{2}}$
- Gradient Orientation — $\theta = \arctan2(G_{y}, G_{x}) \times (\frac{180}{\pi})$

Connected - Component Analysis
- This is used to find connected components in an image using graph theory.
- It helps in finding blobs in binary and thresholded images
- Connected-component labeling (also known as connected-component analysis, blob extraction, or region labeling) is an algorithmic application of graph theory that is used to determine the connectivity of "blob"-like regions in a binary image.
- The actual algorithm consists of two passes. In the first pass, the algorithm loops over each individual pixel. For each center pixel p, the west and north pixels are checked
  - 4-connectivity (left). Based on the west and north pixel labels, a label is assigned to the current center pixel p
  - 8-connectivity can also be performed by checking the west, north-west, north, and north-east pixels (right).

Object Detection Framework
- #1 Experiment Preparation
- #2 Feature Extraction
- #3 Detector Training
- #4 Non-Maxima Suppression
- #5 Hard Negative mining
- #6 Detection Re-Training

Color Spaces
- RGB
  - The RGB color space is the most common color space in computer vision. It's an additive color space, where colors are defined based on combining values of red, green, and blue. While quite simple, the RGB color space is unfortunately unintuitive for defining colors as it's hard to pinpoint exactly how much red, green, and blue compose a certain color — imagine looking a specific region of a photo and trying to identify how much red, green, and blue there is using only your naked eye!
  - RGB color space for most computer vision applications.
- HSV
  - The HSV color space is also intuitive, as it allows us to define colors along a cylinder rather than a RGB cube, The HSV color space also gives lightness/whiteness its own separate dimension, making it easier to define shades of color.
  - if you are interested in tracking an object in an image based on its color. It's very easy to define color ranges using HSV.
- LAB
  - L*a*b* provides with perceptual uniformity, meaning that the distance between two arbitrary colors has actual meaning. However, both the RGB and HSV color spaces fail to mimic the way humans perceive color — there is no way to mathematically define two arbitrary colors are using the RGB and HSV models. And that's exactly why the L*a*b* color space was developed.
  - when you're concerned with color management, color transfer, or color consistency across multiple devices, the L*a*b* color space will be your best friend. It also makes for an excellent color image descriptor.
- GrayScale
  - We often use grayscale representations of an image when color is not important — this allows us to conserve memory and be more computationally efficient.
  - https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/

RGB to Grayscale Conversion
- The lightness method averages the most prominent and least prominent colors: (max(R, G, B) + min(R, G, B)) / 2.
- The average method simply averages the values: (R + G + B) / 3.
- The luminosity method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. We're more sensitive to green than other colors, so green is weighted most heavily. The formula for luminosity is 0.21 R + 0.72 G + 0.07 B.

Contours Applications
- Centroid/Center of Mass
- Area
- Perimeter
- Bounding boxes — bounding box area = bounding box width x bounding box height
- Rotated bounding boxes
- Minimum enclosing circles
- Fitting an ellipse
- aspect ratio
  - aspect ratio = image width / image height
  - Shapes with an aspect ratio < 1 have a height that is greater than the width — these shapes will appear to be more "tall" and elongated.
  - shapes with an aspect ratio > 1 have a width that is greater than the height.
  - shapes with an aspect ratio = 1 (plus or minus some \epsilon of course), have approximately the same width and height.
- convex hull
  - important aspect of the convex hull that we should discuss is the convexity. Convex curves are curves that appear to "bulge out". If a curve is not bulged out, then we call it a convexity defect.
- extent
  - extent = shape area / bounding box area
  - In all cases the extent will be less than 1 — this is because the number of pixels inside the contour cannot possibly be larger the number of pixels in the bounding box of the shape.
- solidity
  - solidity = contour area / convex hull area
  - it's not possible to have a solidity value greater than 1. The number of pixels inside a shape cannot possibly outnumber the number of pixels in the convex hull, because, by definition, the convex hull is the smallest possible set of pixels enclosing the shape.

Object Detection
- Haar Cascade / Viola-Jones
- keypoints
- local invariant descriptors
- bag of visual words
- HOG with SVM
- deformable parts models
- Deep Learning with Pyramids

HOG Descriptors
- The two most important parameters to consider in our HOG descriptor are pixels_per_cell and cells_per_block