Data Set Creator from Image Features

In [1]:
```
from imutils import paths
from preprocessing import ImageResizer, FeatureExtraction
from preprocessing import SimpleDatasetLoader
```

Initialize Objects

- Listing all the images inside the resources directory
- Initializing Image Resizer that will resize all image to specified size
- Initializing feature extraction methods that will extract the features from the images
- Initializing data loader that will load all images from resouces and do preprocessing before it

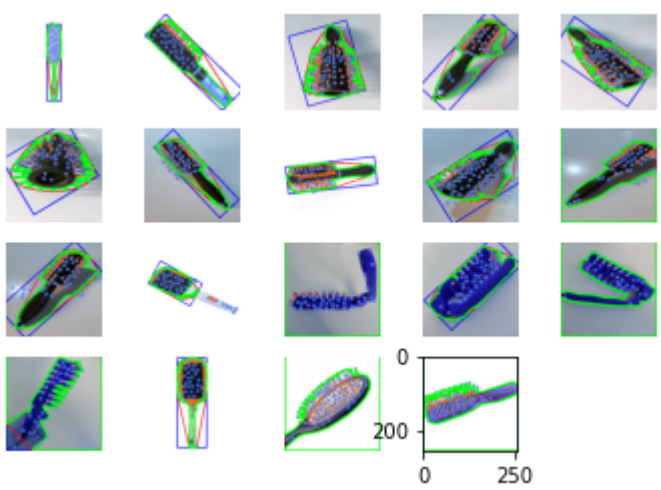
In [2]:
```
# listing all images inside the resources!
imagePaths = list(paths.list_images("resources/"))
# initialize the image preprocessor, load the dataset from disk, and reshape the data matrix
sp = ImageResizer(256, 256)
# initialize the feature extractor that will save it to csv file
f_ext = FeatureExtraction()
# initialize data set loader that will load the images and do the preprocessing before it
sdl = SimpleDatasetLoader(preprocessors=[sp, f_ext])
```

Loading and Extraction Features and Saving the Data to a CSV File

The result of the load is still the images and the labels of the images, you can later for example get the data and still manipulate it. You can also directly use the data and labels for the AI-Processes. The labels will also be used to in the function extract to table. Or you can also generate Panda data frame from the feature extraction.

In [3]:
```
print("[INFO] loading images...")
# load the images
(data, labels) = sdl.load(imagePaths, verbose=100, show=20)
# extract to the csv file
```

```
[INFO] loading images...
[INFO] processed 100/800
[INFO] processed 200/800
[INFO] processed 300/800
[INFO] processed 400/800
[INFO] processed 500/800
[INFO] processed 600/800
[INFO] processed 700/800
[INFO] processed 800/800
```



In [4]:
```
f_ext.extract_to_table("features.csv", labels)
# extraction direct to panda data frame
df = f_ext.extract_to_panda(labels)
df.sample(20)
```

Out[4]:

	n_corner	n_h_corner	n_contour	a_rect	a_hull	a_approx	l_perimeters	wide/length	perim/a_rect	perim/a_hull	...	corner/a_hull
739	36	302	9	10511.996864	9229.0	7330.5	598.710677	4.562500	0.056955	0.064873	...	0.003901
503	33	235	2	34935.000000	34576.5	32228.0	817.053823	1.861314	0.023388	0.023630	...	0.000954
328	84	451	6	26764.457290	20324.5	16509.0	961.602155	0.266633	0.035928	0.047312	...	0.004133
666	80	485	47	65025.000000	63228.0	42810.0	2214.915863	1.000000	0.034063	0.035031	...	0.001265
383	39	628	8	10296.000000	7599.0	5625.5	682.835567	5.318182	0.066320	0.089859	...	0.005132
324	100	591	3	29560.408065	22551.0	17908.0	893.200134	3.200310	0.030216	0.039608	...	0.004434
557	35	289	2	6110.000000	5798.5	5668.0	509.941125	9.038462	0.083460	0.087944	...	0.006036
549	34	357	5	32031.117456	30417.0	29590.5	753.112698	0.471353	0.023512	0.024760	...	0.001118
54	100	807	14	44483.047176	32469.5	26836.5	1764.104681	0.872416	0.039658	0.054331	...	0.003080
762	29	123	2	9881.000000	9167.5	8854.0	562.953318	5.878049	0.056973	0.061408	...	0.003163
211	50	264	2	18297.949064	14442.0	12087.5	682.991983	4.482075	0.037326	0.047292	...	0.003462
721	90	777	6	6780.949348	6169.5	5031.0	495.882250	0.137475	0.073129	0.080376	...	0.014588
645	78	215	2	65025.000000	65025.0	54494.0	1792.589962	1.000000	0.027568	0.027568	...	0.001200
182	38	55	2	64770.000000	64319.5	64231.5	1014.485281	0.996078	0.015663	0.015773	...	0.000591
618	29	208	1	65025.000000	65025.0	56802.0	1599.663994	1.000000	0.024601	0.024601	...	0.000446
757	37	316	5	36411.000000	35009.5	34341.0	763.320849	1.440252	0.020964	0.021803	...	0.001057
708	39	331	2	7424.292748	6752.0	6543.0	479.053823	6.058113	0.064525	0.070950	...	0.005776
28	26	305	1	65025.000000	65025.0	53643.0	1665.754395	1.000000	0.025617	0.025617	...	0.000400
494	100	105	13	8745.000000	6984.5	5934.5	543.438599	0.321212	0.062143	0.077806	...	0.014317
723	52	137	2	31460.000000	31056.5	18556.5	2667.144223	1.861538	0.084779	0.085880	...	0.001674

20 rows × 22 columns