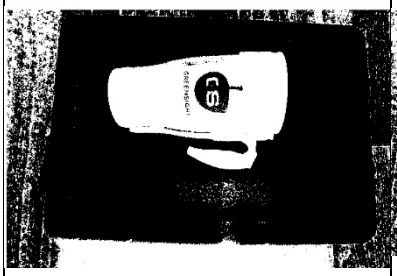
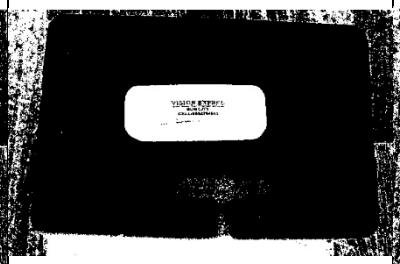
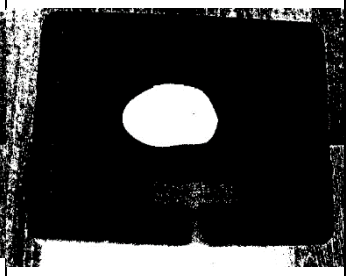
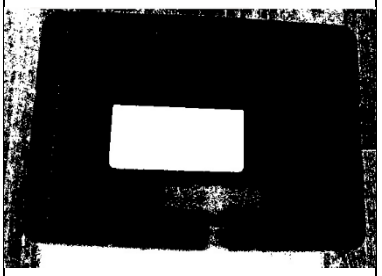
Home_Work-3: Sriram Pandi & Srinivas Peri

1.) Original Images

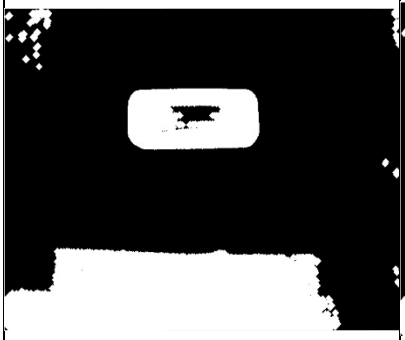
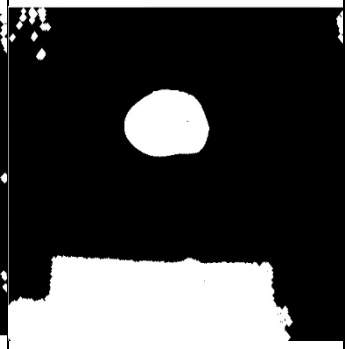
			
MUG	BOX	MOUSE	POWER BANK

2.) Threshold Image outputs:

				
MUG	BOX	MOUSE		POWER BANK

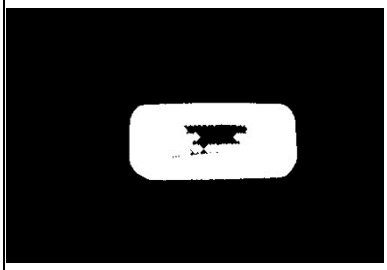
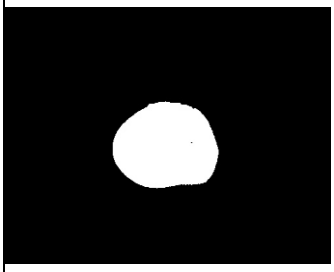
3.) Cleaned Image:

We used the nearest neighbor and KNN algorithms to implement real-time object recognition in our project. If the objects are large enough and far enough away from the border, the system can recognize and classify them all at once. Furthermore, the system allows users to save features from the window's primary component as either training or testing data. These data sets can be used for object classification and evaluation.

			
MUG	BOX	MOUSE	POWER BANK

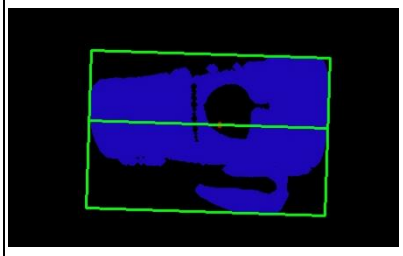
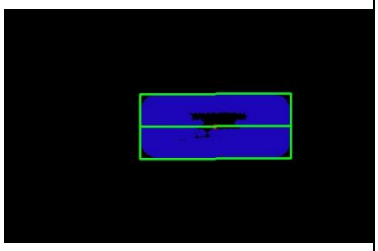
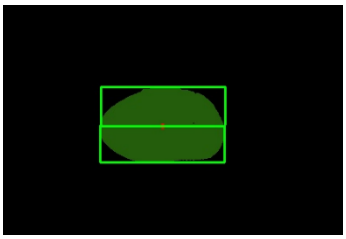
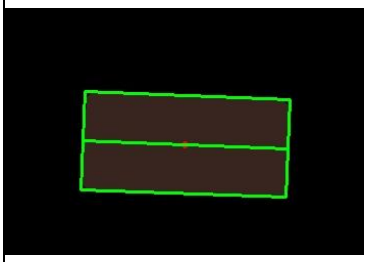
4.) Segmented Image:

The system we created can exclude regions that are smaller than a specified "min area" and only show the top "n" regions. Furthermore, we have already removed regions located near the border.

			
MUG	BOX	MOUSE	POWER BANK

5.) Feature extraction of images:

The moments section of the code computes the centroid, angle, bounding box dimensions, filled ratio, second moment about the central axis, and Hu moments of an image.The "Moments" object returned by cv::moments() represents various image moments, including the 0th moment (the sum of pixel intensities in the image), 1st order moments (which give the image's centroid), and 2nd order moments (which describe the spread of the image). The code then uses these moments to compute the image's centroid, angle, and major axis. The image is then transformed so that its major axis is parallel to the x-axis.

			
MUG	BOX	MOUSE	POWER BANK

6.) **NN Images:**

We used the nearest neighbor algorithm with scaled Euclidean distance for the classifier. Several sets of training data were created and formatted in the same way as shown in the "features.txt" file.

powerbank:0.515403,0.958621,3454.81,0.20458,0.0142685,1.26219e-05,

powerbank:0.508519,0.960303,1587.35,0.205159,0.0146119,8.4978e-06,

powerbank:0.509701,0.970302,2941.08,0.205195,0.014543,8.52897e-06,

mouse:0.679017,0.739687,1029.84,0.17572,0.00494768,6.129e-05,

mouse:0.654279,0.78913,1638.18,0.174953,0.00491038,7.15374e-05,

mouse:0.707902,0.741139,1536.17,0.172975,0.00403135,3.96552e-05,

mug:0.573433,0.664018,6962.43,0.254049,0.0242595,0.000748307,

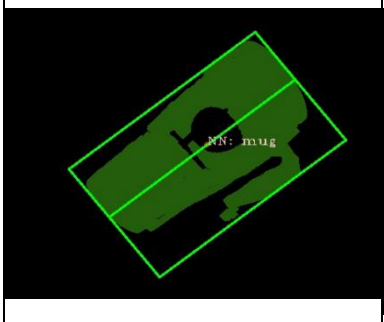
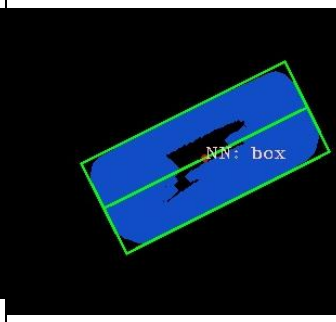
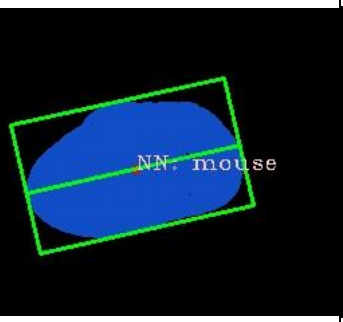
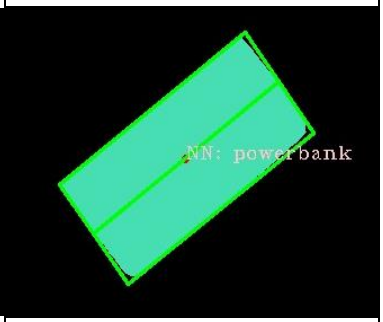
mug:0.761024,0.565825,6306.73,0.258493,0.0123914,0.00553165,

mug:0.800421,0.529038,3346.83,0.258898,0.0126748,0.00585324,

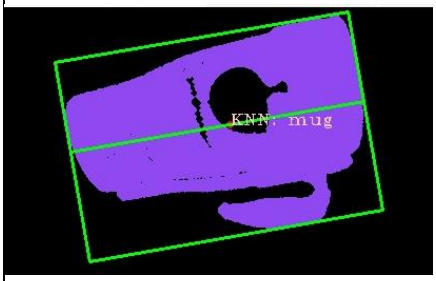
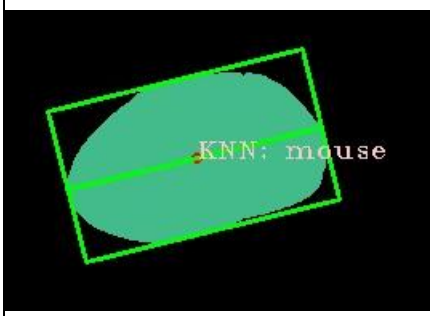
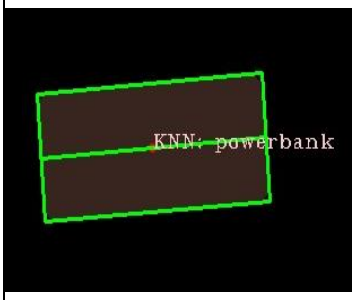
box:0.440958,0.883206,3865.24,0.250915,0.0296657,2.52679e-05,

box:0.440842,0.847331,3557.8,0.243856,0.0258794,0.000117357,

box:0.427536,0.89661,717.763,0.251054,0.0303338,4.38966e-05,

			
MUG	BOX	MOUSE	POWER BANK

7.) KNN Images:

			
MUG	BOX	MOUSE	POWER BANK

Training dataset:

mug:0.724723,0.46668,3281.86,0.257844,0.0141595,0.00690108,

mug:0.797493,0.531499,6041.19,0.251679,0.0118986,0.00430788,

mug:0.818141,0.515219,4085.42,0.246643,0.0118285,0.00344814,

mouse:0.604984,0.786348,1629.76,0.181067,0.00710577,8.26761e-05,

mouse:0.693147,0.746882,1636.35,0.173817,0.00440771,5.70447e-05,

mouse:0.632377,0.756049,597.224,0.181391,0.00697134,0.000109097,

powerbank:0.484179,0.979749,1406.4,0.210919,0.0169655,6.01525e-08,

powerbank:0.488196,0.970246,1485.11,0.211686,0.0172597,3.55054e-06,

powerbank:0.51911,0.962205,2014.74,0.203767,0.0139436,1.59755e-05,

box:0.414429,0.895018,3775.05,0.251085,0.0319223,3.41548e-05,

box:0.460054,0.870252,3854.67,0.240726,0.0238347,4.3692e-05,

box:0.606633,0.714965,2814.83,0.224128,0.0201319,0.000525346,

Test dataset:

mug:0.562749,0.727719,6534.1,0.220508,0.0180136,0.00111102,

mug:0.51873,0.651991,1481.37,0.245962,0.0277129,0.00204863,

mug:0.706204,0.653359,2266.92,0.205653,0.00899956,0.00222637,

powerbank:0.46231,0.892424,3789.52,0.222019,0.0216712,1.26223e-06,

powerbank:0.501411,0.934617,2711.61,0.209397,0.016213,2.46319e-05,

Powerbank:0.396563,0.740417,1401.43,0.321125,0.0587442,0.00114152,

EarPods: 0.632377,0.747549,3295.324, 0.960231,0.0297134, 3.363431e-05,

mouse:0.601854,0.751767,1870.33,0.186801,0.00843655,0.00014462,

mouse:0.62493,0.777253,1754.7,0.178723,0.006234,7.87282e-05,

mouse:0.583362,0.791383,1569.96,0.183697,0.00801202,7.17631e-05,

box:0.436819,0.797285,4250.01,0.279023,0.0359411,5.2976e-05,

box:0.372455,0.801029,630.901,0.270041,0.0386728,0.000294208,

8.) Confusion matrix of test dataset :

	Mug	Box	Mouse	EarPods	Power bank
Mug	3	0	0	0	0
Box	0	3	0	0	0
Mouse	0	0	2	0	0
EarPods	0	0	1	0	0
Power bank	0	0	0	0	3

9.) Link to the working video:

https://drive.google.com/file/d/1ugSMoLBAD2nmhB8U0mDWrcEkVr-9JRxj/view?usp=sharing

summary:

In this project, we used the nearest neighbor and KNN algorithms to enable real-time object recognition. The system can detect and categorize multiple objects at the same time, if the object is large and not close to the edge. Furthermore, users can save features from the window's primary component as either training or testing data. These data sets can then be used for classification and evaluation purposes.

Reflection:

This project taught us how to use features in content-based image retravel and how the object detection back-end works. The previous exam questions helped us understand the project's work flow, which enabled us to implement the grassfire transform and learn many new concepts such as cv::moments and features. And how to use morphological operations to improve the detection process.

References and Acknowledgement:

https://www.gormanalysis.com/blog/reading-and-writing-csv-files-with-cpp/

https://youtu.be/yxG2df04YJk

https://docs.opencv.org/3.4/d0/d49/tutorial_moments.html

https://github.com/ulikoehler/cv_algorithms/blob/master/doc/Grassfire.md