Criterion B: Design

Break Down:
The following describes the methods in each script or class and what each method does.
Tool.cpp

Name	**Parameters**	**Functionality**
SplitString	string text, string splitters	It splits the string text at one of the splitter chars in splitters.
TrimEnds	string text, string chars	Trims the front and back of text if they are equal to one of the chars.

ConfigReader.cpp

Name	**Parameters**	**Functionality**
MakeConfigFile		Make a new config file with the date in the program.
RemoveConfig		Deletes the config file.
ReadConfig		Reads all the stored values from the config file.
GetConfigVarsFromID	int id	After the ReadConfig method is called this method returns the config value assigned to the id set by the enum ConfigVarsID.

Trainer.cpp

Name	**Parameters**	**Functionality**
Train	String arg	This is called to start the training algorithm.
CorrelateImageFeatures	list<feature>* keyFeatures	Checks to see if there are any duplicates of features like a car having two wheels and adds them together.
GetInputImages	list<string>* imageFiles	Gets n number of training image file paths from user.

GetMostImportantPartsOfImage	Mat *grayImage, int maxFeatures, float xStepSizePersentOfImage, float thresholdPresent, int maxfeatureSizeInSteps, int minfeatureSizeInSteps	Essentially just gets the Most Important/Complex Parts Of the image.
UseRating	const feature& first, const feature& second	This is used with the built in sort function to get each feature's "rating" variable.
CorrelateFeaturesCrossImage	list<list<feature>>* featuresForEachImage	Matches keywords from different training images.
CreateKeywords	list<list<pair<feature*, Point>>> keywordLists, int thresholdOfSharedImages	Adds every image together according to their offset in each sublist returning a list of Mats.
SaveMultipleAsImgFiles	string folderPath, string folderName, list<feature>* images	Saves multiple keywords to file.
SaveOneAsImgFile	string folderPath, string folderName, feature* image	Saves one keyword to file.

OpenCVTools.cpp

Name	**Parameters**	**Functionality**
LoadImages	list<string> paths, ImreadModes im	Returns a list of images from the list of file paths, paths.
ShowImages	list<Mat>* imgs	Shows a list of images using the built in OpenGL imshow method.
ShowImage	Mat* img	Shows a single image using the built in OpenGL imshow method
GetGradientImage	Mat grayScaleImage	Uses the Sobel operator to make an edge image.

MakeMatFromRange	Point start, Point end, Mat* image, bool ShowImages	Makes a Mat from a range of another Mat.
DoOneCorrelation	Mat *image, Mat *templ, Point p, Mat *result	Gets the correlation of two images at a given offset, p, taking into account the amount of overlap.
DoesCorrelationReachThreshold	Mat image, Mat templ, float threshold, Point *location, bool scaleImg2ToMatchRows	Gets the location of the best correlation and returns if the correlation passes a threshold.
AddImagesAt	Mat* img1, Mat* img2, Mat* result, Point offSet, Point* mainImageOffSet, float ratio, bool cropEdgesToSquare	Adds two images together at an offset, offset.
TotalMatAddByOne	Mat image	Increments every point in the image by 1. This is to make 0 represent transparent because there is no value for this in a uchar (byte).
CorrelateWithConvolution	Mat src, Mat templ	Uses the matchTemplate method to get the best location for a match.
HoughFeatureAccumulater	list<feature> features, Mat src, float PresentOfBinSizeForOutput, int removeRadius, bool ShowImages	Gets all the data associated with where the center or centers could be in the image.
UseValue	const pair<float, Point>& first, const pair<float, Point>& second	This is used with the built in sort function to get the pair's first value.

FindPeaks	Mat image, int numOfPoints, float peakAmountThreshold, int removeRadius	Find the spots in a Mat with the highest values.
DrawPointOnImage	Mat src, Point p, Scalar color, int size	Draws a point at point p on the image

Detector.cpp

Name	**Parameters**	**Functionality**
Find	string arg	This is called to start the detection algorithm.
GetObjectsToFind		Gets n number of objects to find, from the user.
GetFilePathsFromObjectNames	list<string> objects	Looks at saved keyword data using the names the keywords were saved as.
CreateFeaturesFromFolderPath	string folderPath	Reads all keywords from the file and puts them into the feature class.
DrawOnImage	Mat src, float PresentOfBinSizeForOutput, list<pair<float, Point>> peaks	Draws on the image at each point in peaks.

Source.cpp

Name	**Parameters**	**Functionality**
main	int argNums, char** argv	Directs what algorithm is to be used, training or detection, based on the argv.

Tools.h feature class

Name	**Parameters**	**Functionality**
GetRating		Rates how complex the image is using the gradient of the grayscale image.
operator==	const feature& feature1	If something like "(feature1 == feature2)" is used, it returns true or false based on whether or not the grayScale images are equal.

Test plan

To verify that the program works as intended, test images will be used. Three test images of the same object will be fed into the trainer, and the resulting keywords will be used to find where the object is in the third test image. Tests to be performed include:

- Verify that the program is capable of creating useful “keywords.”
- Verify it can store and retrieve keyword files.
- Verify that it can find possible locations of the object.
- Finally, verify that it can draw on the image to indicate the position of the object.

Words 93