2020

# CITS4402 Computer Vision Project – Group 18

Object Recognition in Cluttered Scenes

DAVID STEKETEE (21962175)
JOSHUA POLLARD (21966798)
THOMAS HILL ALMEIDA (21963144)

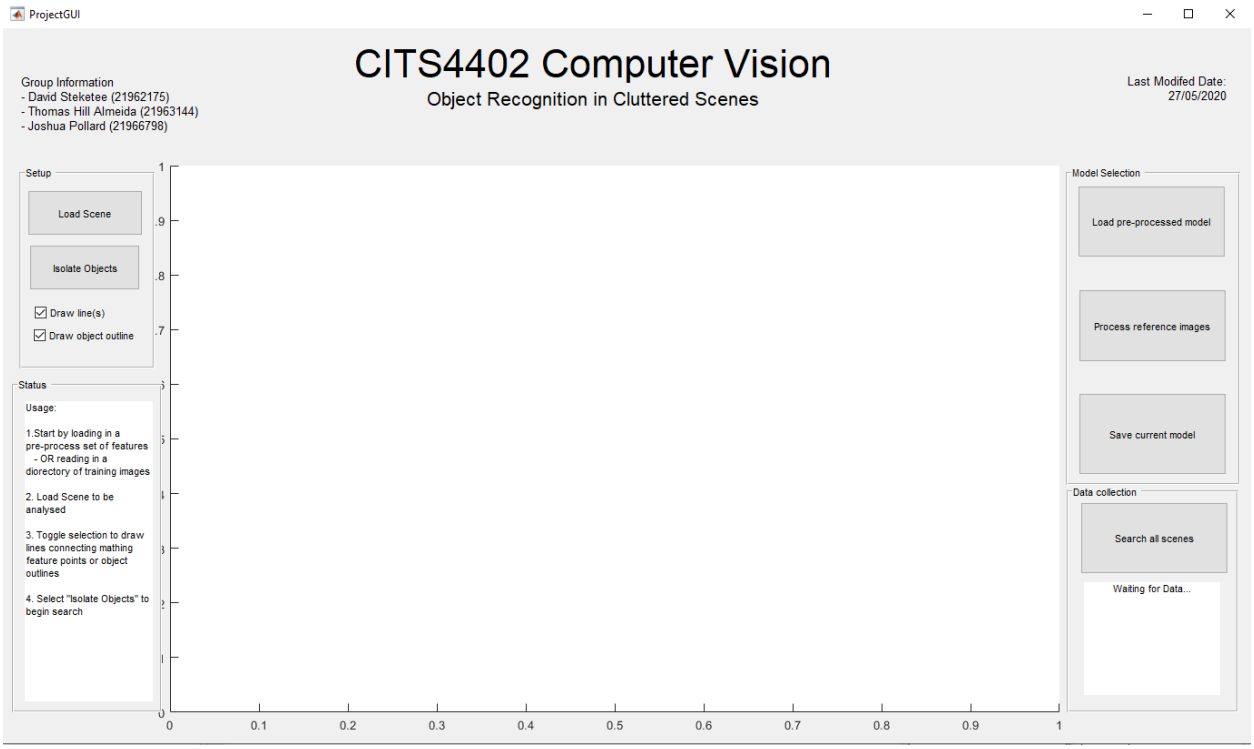
# Contents

# Overview

In this project we have built an object recognition system in MATLAB using SURF features to recognize and locate reference images in a cluttered scene. 20 different reference objects were placed in various scenes. The system was able to detect objects present in these images with reasonable accuracy even with significant occlusion and clutter in the scene images. 10 Scenes images were taken for 3 difficulty levels – easy, difficult and very difficult. Easy scene images were those with no occlusion of target objects. Difficult scenes had some occlusion of target objects. Very difficult images have significant occlusion of target objects and clutter in the image.

# Usage guide and explanation

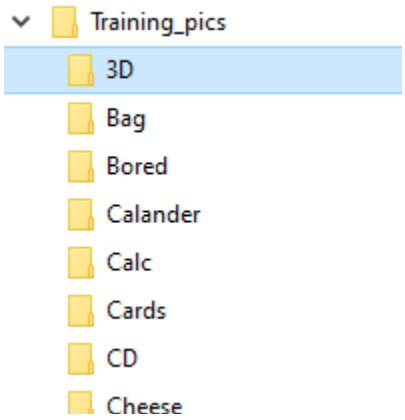
The entire object recognition system can be trained and operated from a single graphical user interface. When the GUI is first loaded it will look like this.




There are basic usage instructions that can be seen in the bottom left corner panel labelled ‘Status’.

To get started:

1. Select ‘Process reference images’ on the right and select a directory of object reference images.
    - The directory selects must have a folder inside of it that holds a number of images of the reference object. Example shown below




- The name of these folders will be extracted as the name of the object.
- The SURF features of each image with be stored in a structure variable along with some meta data such as the name of the image file
    i. During this process the background is automatically filtered out from the reference images before SURF features are calculated. And a convex hull is formed that approximated the outline of the object. This convex hull is saved as a mask along with the meta data
- When system first starts to process the training images the message “loading training images...” will be displayed in the status box. Once the processing has finished the message “training images loaded” will be displayed

2. Select ‘Save current model’ on the right to save the SURF features structure as a .mat file
    - Previously saved structures can be loaded using the ‘Load pre-processed model’ button on the right to load any previously saved structures in .mat files
        i. A ‘full_set_feats.mat’ file has be included as well as a ‘quick_run_feats.mat’ file for quick visualization
    - Message displaying that the model has been saved will be displayed in the status box
3. To identify objects in a specific scene start by selecting the ‘Load Scene’ button in the top left and select a scene image to analyse.
    - This scene image will be shown I the main axes
4. Toggle check boxes to select if lines should be connected between SURF features in scene and reference images and/or if object outlines should be drawn
5. Select ‘Isolate objects’ button on left.
    - Status box will display that search process has started.
    - System extracts SURF feature points from scene image then for each object compares the feature points and finds matches
    - System then attempts to find an affine transformation that maps a set of the reference image feature points found to scene image feature points within a tolerance of 7 pixels. Set of points that are mapped by this transformation are stored
    - System check that the transform found works for at least 8 unique sets of points and that points in the scene and reference image are between 100 and 30000 pixels apart. This filters out weak transforms found or transform that dramatically enlarge or reduce the size of the object during mapping with are incorrect as all objects in reference and scene images do not vary that dramatically in scale (this prevent a common problem of

incorrectly mapping all feature points in one image to a single point of very tight clump of points in the other image)

6. Reference images will be appended to scene image on right hand side of axes as they are found and connecting lines and object outlines drawn if specified by checkboxes prior to run time
7. To run through all 30 scene images the 'Search all scenes' options can be selected. User can then select a directory of folders for each difficulty of scene images to identify.
    - The system will run through each directory keeping track of all images found in each scene
    - Once all scene images have been processed the objects found are compared against the known objects in the scene (stored in text files in main directory) and report the number of object correctly located and the number of false positives.
    - This information will be displayed in the 'Data collection' box in the bottom right

## Quick testing guide

A pre-evaluated set of SURF feature points for our full 20 reference object has been provided but this unfortunately still take a few minutes to load and evaluation of each scene image takes approximately 30-40 seconds. If you would like to view functionality of this system quickly a 'quick_run_pics' directory has also been provided with only a few reference images and scenes. Please follow the steps above using the 'quick_run_pics' directory instead of the 'full_set_pics' directory to quickly see the system in operation and to test the functionality of the 'Search all scenes' button without waiting for the 20-25 minute processing time.

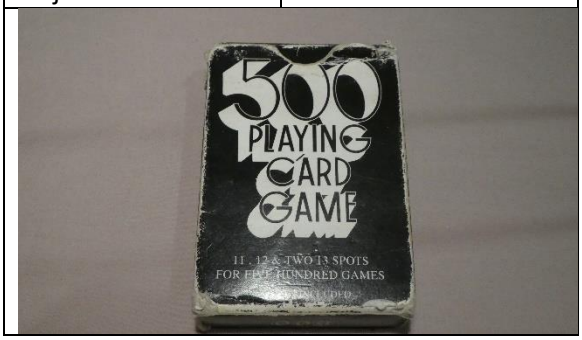
## Reference objects

20 different reference objects were collected to be search for in scene images. 6 photos of each object was taken from different angles making a total of 120 images. One each of each object is shown below

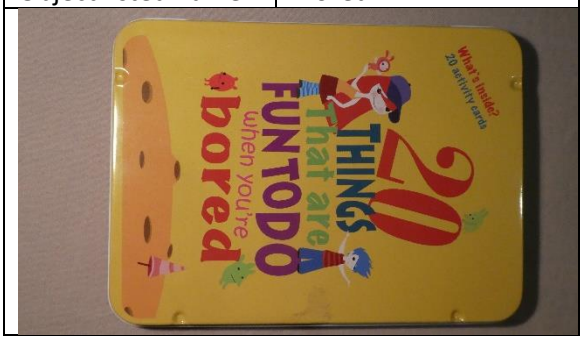
| Object: | Pairs of 3D glasses |
|---|---|
| Object listed name: | '3D' |

| Object: | Paper bag with repeating geometric pattern |
|---|---|
| Object listed name: | 'Bag' |

| Object: | Deck of playing cards |
|---|---|
| Object listed name: | 'Cards' |

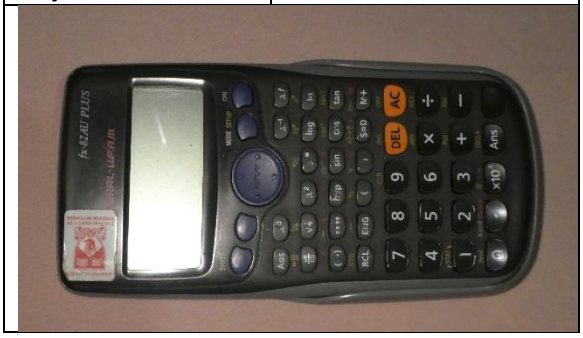



| Object: | Tin box labeled "20 things that are fun todo when you're bored" |
|---|---|
| Object listed name: | 'Bored' |

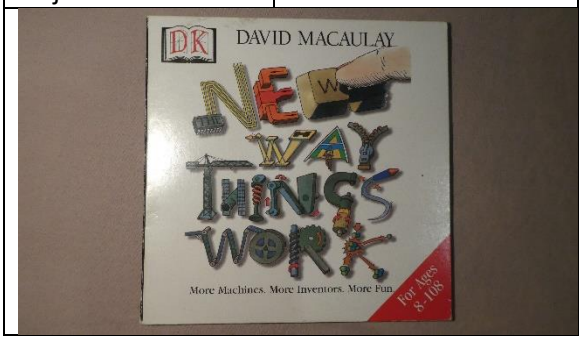



| Object: | Metal and wooden rotating Calendar Dial |
|---|---|
| Object listed name: | 'Calander' |

| Object: | Scientific calculator |
|---|---|
| Object listed name: | 'Calc' |



| Object: | CD case labelled “New way things work” |
|---|---|
| Object listed name: | ‘CD’ |




| Object: | Disney themed card holder keyring |
|---|---|
| Object listed name: | ‘Disney’ |


| Object: | Packet of plastic toy cheese |
|---|---|
| Object listed name: | ‘Cheese’ |


| Object: | Three pointed highlighter |
|---|---|
| Object listed name: | ‘Highlighter’ |


| Object: | Brown hair comb |
|---|---|
| Object listed name: | ‘Comb’ |

| Object: | Mario tennis Gameboy advance box |
|---|---|
| Object listed name: | ‘Mario’ |

| Object: | Box for micro:bit go |
|---|---|
| Object listed name: | ‘Micro’ |

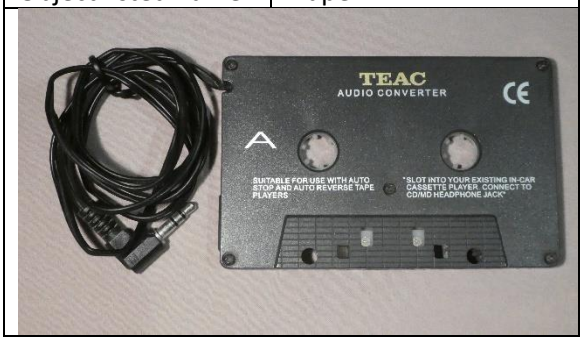

| Object: | Green plastic three blade rotor |
|---|---|
| Object listed name: | ‘Rotor’ |

| Object: | Multitool |
|---|---|
| Object listed name: | ‘Multi’ |

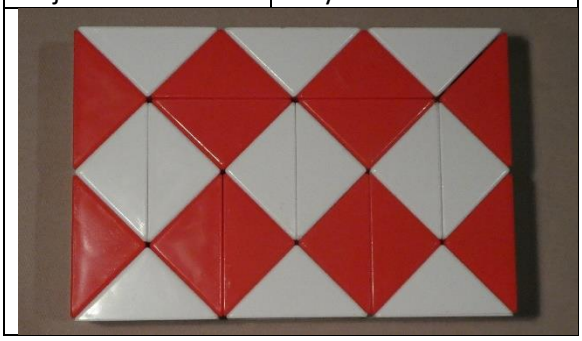
| Object: | Blue UWA 30cm ruler |
|---|---|
| Object listed name: | ‘Ruler’ |


| Object: | Red telephone box themed phone case |
|---|---|
| Object listed name: | ‘Phone’ |


| Object: | Cassette to AUX audio converter |
|---|---|
| Object listed name: | ‘Tape’ |



| Object: | Red and white rotatable rubix toy |
|---|---|
| Object listed name: | ‘Toy’ |

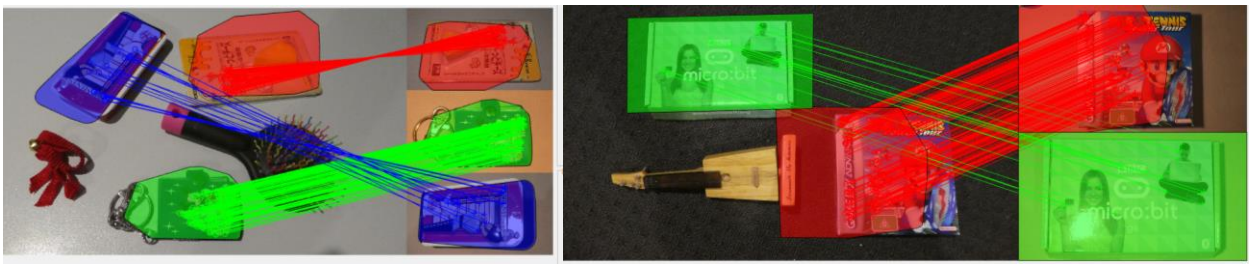


| Object: | Black Rottnest wallet |
|---|---|
| Object listed name: | ‘Wallet’ |


## System Evaluation

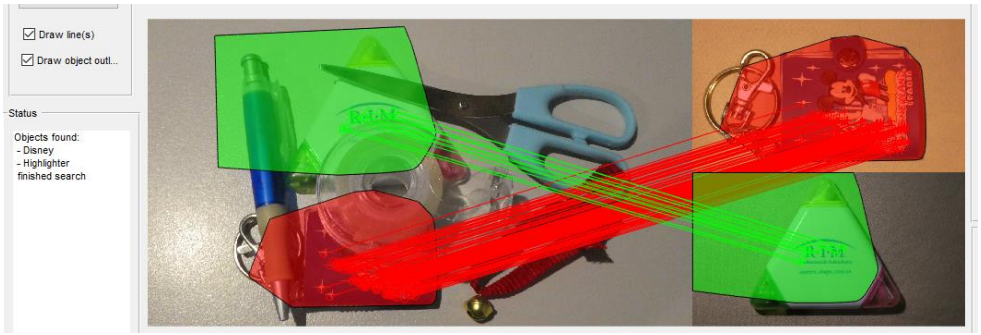
Out system is able to correctly identify object in cluttered scenes are required. The object recognition system is robust to significant occlusion and clutter in some cases. Example outputs shown below

**Line and outline on easy scene**



**Line and outline on difficult scene**

**Line and outline on Very-difficult scene**



Object outlines were created by approximating and filtering out background colour then creating a 'convex hull' around the remaining image. For some background with non-uniform illumination such as the triangular highlighter above this mean that the outline sometimes incorrectly included a larger section of the background. Ideally a more sophisticate system could be used to extract accurate image outlines but as the system already can be quite slow to run due do the large number of SURF features generated and processes it was determined that improving this outline was not a priority. The mapping of the outline from the reference image to the scene image by applying the affine transformation found mapping the corresponding feature points is shown to work relatively well, correctly applying scale and rotating and having a relatively accurate position.

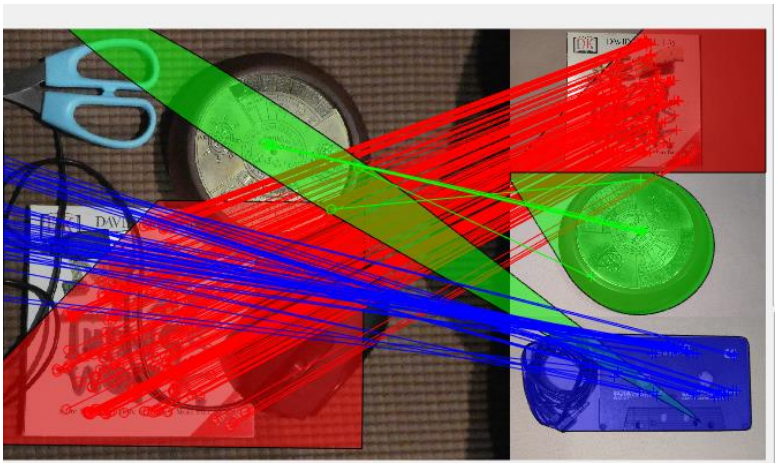
## Results

| Difficulty | Number of correctly identified objects | Number of false positives |
|---|---|---|
| Easy | 17 / 25 | 1 |
| Difficult | 16 / 24 | 2 |
| Very Difficult | 16 / 24 | 0 |

As shown above our system correctly identified 49 out of 73 objects (67%) and only falsely identified objects 3 times across all scene difficulties. We noticed that the objects correctly identified in one image, were almost always correctly identified in all other images they appeared in regardless of difficulty. This shows that the classifier is resilient to occlusion. Likewise, the object not detected in one scene were usually not picked up in any other image. This shows that the feature detected for these objects were not sufficient to identify them, suggesting that the feature extraction was not sufficient for these objects.

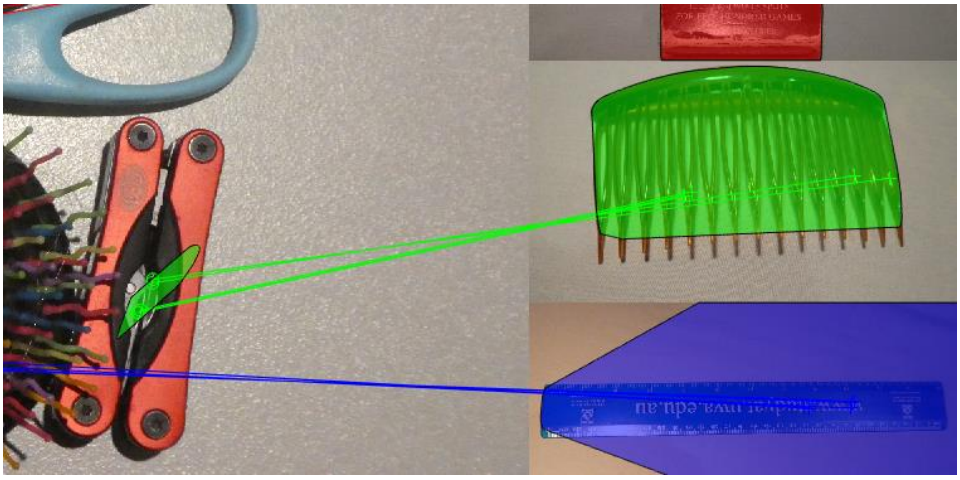
## Future improvements & Lessons learned

We developed a deeper understanding of the complexity of object vision tasks and computer vision in general during this project. We noticed that matched surf features could be used very effectively on objects with a large number of unique corners and patterns of a medium scale especially text and small pictures. But the matching process was very difficult for objects with no pattern or texture, such as the green rotor blade, or with continuous or repeating structure, such as the 'Calander' disk. The calendar disk was often correctly located but the points were not connected accurately so the translation art of the affine transformation found was roughly correct while the shear component of the transform was very wrong as shown below.



Objects with text including the ‘CD’ and the ‘Calc’ or unique picture such as ‘Disney’ and ‘Phone’ were picked up with high accuracy, even for the very difficult scenes, while objects of solid color or few details such as ‘rotor’ and ‘Toy’ were very rarely picked up, if at all, even for the easy scenes. The vast majority of errors made by the classifier were false negatives for these simpler objects, with false positives a rare occurrence. This could be improved by refining the features extracted by collecting features across multiple images of the same object and using a leaning algorithm to extract the key features. Alternatively, a different feature extraction method may be more suitable for plain images that looks at broader details rather than the matching of individual points.

Initially there was a high number of false positives, primarily caused by all features being matched to either a single point or collection of points in an area of high detail which happened to align. Visually these were obvious to notice (see example below) and are now filtered out by enforcing a minimum max distance between points for the detected object.



As seen by the outline shapes we had difficultly distinguishing between the object and the background when the colours were similar, this could be improved by using backgrounds of a distinctly colour to each object, such as a dark background for all light colored objects and light backgrounds for dark objects.

## Conclusion

The classifier produced was able to correctly identify 67% across 30 different scenes, maintaining this accuracy across all 3 difficulty levels, showing that occlusion had little effect on its ability to recognize objects. It also only produced 3 false positives across all 30 scenes, showing a high degree of certainty in its predictions.

There were found to be difficulties caused by the feature extraction across different images, with some images not being recognized in any scene image. We identified this as being caused by the objects lacking clear distinguishing features or a color similarity to the background. This is supported by the highest accuracy being achieved by images with clear features such as text, and a clear distinction from the background of the training images.

This classifier may be improved by using training images that have a distinctly different background. A larger number of images used together with a cross image feature extraction would also greatly improve the results without the need to worry about background details. Finally a different feature matching algorithm that matches broader features rather than points would help to overcome these challenges.