Introduction to Vision and Robotics
Assessed Practical 1: Shape Recognition

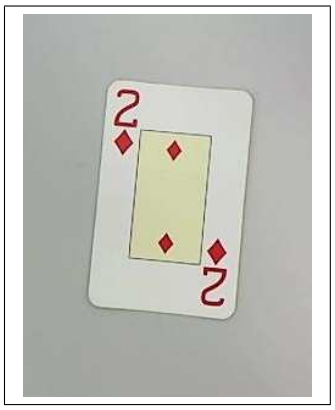
December 12, 2014

1 Introduction

This practical develops a program that detects and identifies playing cards based on images from one of the lab webcams. The program that you develop will:

1. Detect the card region
2. Detect the card features and describe them
3. Identify the card based on the features

Two sample cards are here:

a) 

b) 

You will work in pairs and must submit one joint report *but* this report must be accompanied by a short explanation of how the work was shared and how you think

the (joint) mark should be distributed. 50:50 is OK if you shared the assignment fairly. **You can choose the partner yourself, but you will also have to choose a different partner for the second practical assignment.**

You should be able to normally access the IVR lab (AT 3.01) anytime with your swipe card. The webcams will generally be available in the lab - do not take them elsewhere!

Normally students use Matlab, but you are free to use other software environments. You can use Matlab at any time on other DICE machines, but there are license number limits on availability so *never* stay connected to Matlab when you are not using it.

2 The Task

The overall goal is to develop an algorithm that classifies which card is observed in an image.

Two sets of images are supplied. There are also webcams available in the lab so you can capture additional images of your own cards if you want (details are below).

What you have to do is:

1. Detect the card region: This can be done with thresholding. You might use the `bwmorph(fore,'open',1)` command (with 1 or 2 as the parameter) to remove small noise and the thin black lines. Or maybe you need to use another technique.

2. Detect the card features and describe them: You can find the card regions using `bwlabel(foremm,4)` and `regionprops(labeled,['basic'])` commands. However, you will need to write some heuristic code to decide which 2 of the N+4 regions for the card with value N are the digits and which are the suit symbols.

3. Identify the card based on the features: The lectures presented a Bayes classifier for simple shapes. You might use a combination of colour (*e.g.* the normalised red channel value - see below) and invariant moments to classify each shape. Or you could use colour first to reduce the data to 2 subsets and then use a 2 class Bayes classifier. Or you could use a support vector machine if you have learned about these. Be sure to have enough training data to train your classifiers.

   Then you could use the individual classifications to vote for the card.

You should split your cards into 2 sets, half for training and the other half for testing (the test half was not used when training the classifier).

The **normalised RGB** representation is computed by:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$$

This represents the image values so that they are reasonably independent of the illumination. It also means that white, grey, and black pixels end up with about the same value.

You should produce algorithms that are insensitive to small changes in illumination and camera position.

You should test these aspects of your program.

3 Files You Need

You will need these files (which can be downloaded from the IVR webpage):

Dataset 1: `ivr1415pract1data1.tar` - a set of 32 images

Dataset 2: `ivr1415pract1data2.tar` - a set of 32 images

Each has 32 images named `train1.jpg`, `train2.jpg`, ... `train32.jpg` (or `test1.jpg`, etc), one for each value (2..9) and suit (s=1,h=2,c=3,d=4). There is a ground-truth datafile GT_training.mat that, when loaded, has a 32x2 array of [suit,value] pairs, *i.e.* 8 of clubs is encoded as [3,8].

4 Image Capture Details

You are welcome to use your own cards to produce more training data and example images. Robot 'arenas' (1m x 1m boxes with high white walls) and webcams are available in the IVR lab. You can also capture images of cards on the table tops or the floor.

Use `mplayer tv:// -tv driver=v4l2:width=640:height=480:device=/dev/... video0 -frames 3 -vo jpeg` to capture 3 frames of a new sequence. (Note: this is v4-ell-2, not v4-one-2, as in video-for-linux.) Use the final frame (`00003.jpg`).

5 Writing the report

The report should be a concise description of what you did, why, and what happened. The entire report should be about 5-10 pages including images and tables (excluding

appendices). It should contain the following sections:

1. Introduction: An overview of the main ideas used in your approach.

2. Methods: Explain the vision techniques that you used. Then give a functional outline of how these ideas were implemented and the structure of your code. Explain how each part of it is meant to work. Where suitable, justify your decisions, e.g. why you used one method rather than another, what you tried that didn't work as expected, etc.

3. Results: Show some examples of your results for each stage of the process. Give a performance summary for each stage, showing examples of success and failures. This should include summary performance statistics for: card location, symbol location, symbol recognition, card recognition (*i.e.* what percentage correct for each stage).

   Also, include three confusion matrices. The first shows the classification performance when the cards are grouped by suit (*i.e.* a 4x4 matrix). The second shows the classification performance when the cards are grouped by value (*i.e.* a 9x9 matrix). The third shows the classification performance of the individual suit symbols (*i.e.* a 4x4 matrix), where, for example, the 2 of diamonds card shown in the example above contributes 4 samples to the matrix, and the 8 of clubs contributes 10.

   Well documented failure will get more marks than unsupported claims of success (well-documented success would be even better!).

4. Discussion: Assess the success of your program with regard to the reported results, and explain any limitations, problems or improvements you would make.

5. Code: Any new code that you developed for this assignment. Do not include code that you downloaded from the course web pages or the internet. Any other code that you downloaded should be recorded in the report, but does not need to be included in the appendix.

Your final mark will be based on how well you explain your approach to the task and evaluate the capability of your program as well as the performance.

Submission

Your submission should be a single PDF file and should be submitted electronically by *4pm Thursday Feb 26*. The command to use for on-line submission is:

```
submit ivr 1 FILENAME
```

where `FILENAME` is the name of your PDF file.

The assignment is estimated to take 10 hours coding/test and 5 hours report writing per person, resulting in a c. 6 page report plus an appendix (containing all new code that you wrote, **but not any library or external code that you borrowed**). You must do this assignment in pairs and assign credit in your final report. The assignment will be marked as follows:

Issue	Percentage
Program Design, including comments	25%
Report Clarity	25%
Experimental Results (in Report)	25%
Live demonstration Results	25%

Live Demonstration

There will also be a demonstration session assigned between 9:00-13:00 on Friday February 27, where you will have to demonstrate your code on new data. We'll email you about the location and schedule.

The data will be captured live (by you from sample cards) presented by the marker. You have to write the code to capture the image from one of the lab webcams.

You will need your program to show, for each image:

1. A bounding box around the detected card.
2. A binary image showing the card numbers and suit symbols.
3. A statement of which card it was, *e.g.* "9 of CLUBS".

The live demonstration results will be marked based on a combination of card detection (5%), feature isolation (5%), card recognition (10%), and processing speed (5%).

6 Plagiarism Avoidance Advice

Good Scholarly Practice: please remember the University requirement as regards all assessed work. Details about this can be found at:

www.ed.ac.uk/schools-departments/academic-services/students/postgraduate-taught/discipline/academic-misconduct

and at:

www.inf.ed.ac.uk/admin/ITO/DivisionalGuidelinesPlagiarism.html

You are expected to write the document in your own words. Short quotations (with proper, explicit attribution) are allowed, but the bulk of the submission should be your own work. Use proper citation style for all citations, whether traditional paper resources or web-based materials.

If you use small amounts of code from another student or the web, you must acknowledge the original source and make clear what portions of the code were yours and what were obtained elsewhere. You can ignore this condition for the IVR lecture examples, which can be used freely. The school uses various techniques to detect plagiarism, included automated tools and comparison against on-line repositories. *Remember: a weak assignment is not a ruined career (and may not reduce your final average more than 1%), but getting caught at plagiarism could ruin it.*