Using Image and Video Analysis Techniques to Separate and Identify Dr. Who Trading Cards

University of Warwick

Name:	Oliver Sheldon
Student Number:	2191362

Task 1

First attempt

My first strategy for separating the train-xxx.jpg group of cards was to use the bounding box method introduced to us in lab-2. The separation is achieved in my code using the image_separator method.

My first iteration of the image_separator method used thresholding to separate the cards into separate components, where the skimage.measure.label method could be used to label the components. These components could then be iterated through, and considered as a total card component if there size was between a minimum and maximum size decided by me. Those components that fit this criteria then have the bounding box vertices stored. This list of vertices is then used to crop the cards out of the training images.

With this method I also tried to extract information of each card such as the principle axis and centroid. This would allow me to rotate the cards such that all the card images were upright (or upside down). I found that this method was not successful with the image analysis techniques used in the image_separator method as the principle axes and centroid methods relied on the thresholded image of the cards. This meant that cards that had particularly high contrast between the two sides of the card would return a very inaccurate result for the principal axis and centroid.

One major problem with this method was the amount of manual tuning the method requires. For each training image, I had to manually adjust the scale (the factor that the training image is reduced by in the analysis method), and the gauge (the factor that training image is thresholded by). Despite this, I was very satisfied with the results from this method.

The only image that was not separated correctly was card ID 120 as can be seen in figure 1. This is because of the white background of the card.

Second attempt

I had to return to this task multiple times as I felt my results were not of high enough quality to complete the other tasks to a satisfactory degree.

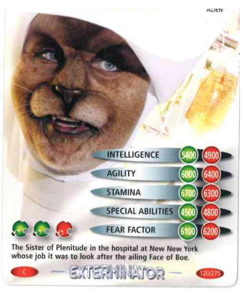
ALIEN
INTELLIGENCE 5400 4900
AGILITY 6000 6400
STAMINA 6700 6300
SPECIAL ABILITIES 4500 4800
FEAR FACTOR 6100 6200
The Sister of Plenitude in the hospital at New New York whose job it was to look after the ailing Face of Boe.
EXTERMINATOR
120/275

Figure 1: Separated card ID 120. The white background has made the thresholding technique unable to correctly crop the card.

For my second attempt, I decided to try to utilize the OpenCV library to achieve my goals. I made the decision to use this method to try to improve my already separated cards from attempt 1.

The main method for this attempt was rectangle_finder to threshold, then find contours, iterate through the contours discriminating based on the contour area using the minimum rectangle area cv2 method. This information could then be used to return the angle subtended by the rectangular card. This angle information could then be used to rotate each card.

I made the decision to use this method on the already separated cards from attempt one. On reflection, and as I found in task-4, with proper image analysis techniques to prepare images, this method can be used on the train-xxx.jpg images with good results, and would not require the manual tuning of parameters.

The rotated card images were then cropped in various ways. I tried to write a method that would automatically crop the cards to remove the white background, but I found this ineffective. In response to task-3 where same size images are required for the CNN I created the crude_auto_cropper method that would crop each card image to the same size. This meant inevitably losing some information on the sides of the card, but the discriminating information for each card was held mainly in the image of the character for the card, so the loss of information from the sides did not concern me.

Task 2

I initially started this task using methods described in lab-6. I created a method matching images using colour histograms but found the results very poor.

I used the sift.detectAndCompute method from OpenCV to create keypoints and description list for the cards separated from training images created in task-1. This is performed by calling the fast_flanner method. This method creates a keypoint and description list for the target image, then uses knnMatch to match these lists to the already created keypoint and description lists for the cards created from task-1. The fast_flanner method returns a list of numbers representing the index of the images with closest matches.

I qualitatively compared the speed of different parameters used in my approach. This included reducing the size of the target image and task-1 imageset, blurring and cropping out non-discriminating information from the image. Cropping and resizing significantly improved the speed of the card identification, and I found that reducing the size of the card image by 90% still created accurate image classification.

This method was very successful at accurately matching images. The method is also quite slow, even after my attempts at optimization through the SDAC method to move a lot of the processing to a one-time initialization stage. One potential improvement that would be easy to implement would be to only find the keypoints and descriptions of a single card from the training images for each ID. This would reduce the number of comparisons by around half, theoretically doubling the speed of my algorithm.

Task 3

I designed a convolutional neural network (CNN) to complete this task. However, before attempting this task I had to modify task-1 to create images of the cards that were all of the same size. CNN's also do not account for rotation or scale, therefore I had to make a decision: Artificially increase the training data with rotated versions of each image or process images to be rotated correctly with respect to the data set. I began with the first suggestion, however I am sure that in this data set the second approach is superior for speed and efficiency as rotating every image in your training set increases the resources used by the CNN by a large amount. There is also an ambiguity to how many rotations of each card are used to train your CNN.

I have no previous experience with machine learning, so I did not proceed far in my predictor. Analysing the final predictor, I know that it is not useful for predicting card ID numbers for the unknown cards extracted from training-images001.

Task 4

I chose to implement the methods devised in task-2 in this task.

In my first attempt I tried to remove the green using a method like chroma key, however I could not find parameters for green values that created satisfactory results. In my second attempt I tried to use a harris corner detector approach, but found this to be not useful for these images, and also very slow. My solution came from a difference of gaussians approach, which turned the contrast-variant green background into a very contrast-invariant (uniform) light blue. I could then easily reuse my chroma key method on this new image. One downside with this method is the difference of gaussians operation is quite resource intensive, so many not be applicable to a real time card identification method.

With the background removed, I reused the rectangle_finder method I developed for task-1 with slight modifications to find the cards, and cropped a square around the card so that no matter the rotation, the whole card would be recovered. This image was then matched with the cut out training cards using SIFT. The SIFT search method (frame_labeller) is far slower than the method to 'cut out' the cards (I called cut_out_cards_rectangle_method).

I tried to improve the speed of my algorithm so that it could be applied in real-time, but found the frame_labeller function to be far too slow to achieve this. One method I would have implemented given more time would be an optical flow estimation, so that once a card had been identified and labelled, subsequent frames would require no image matching or card finding work.

The results of my methods can be seen in the results videos included in my submission. When the camera moves quickly, the blurring affect makes the card identification unreliable but still successful some of the time. The inclusion of cards not included in the training images made the card identification quickly change between seemingly random ID's. This is not unhelpful as this behaviour indicates that the image is not known by the program. One thing my program does not account for is problems of distortion from homography. Despite this I found this to not have a very large affect. If the angle of elevation for the camera was lower, I suspect my method would no longer work. My program also does not indicate how certain it is about the a card identification, and would not be very difficult to include. If a card is not fully shown within the frame, my program also does not identify it due to the rectangle_finder method.

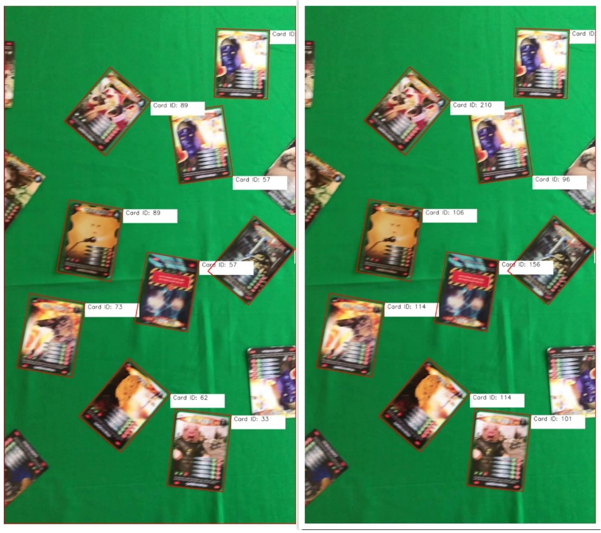

Figure 2: Frame 100 of video1 after labelling with image resize factor 0.2 on the left and 0.1 on the right. Unlike in task 2, due to the blurring from the camera movement resize factor 0.1 is identifies incorrectly. Resize factor 0.2 identified 3 of the 7 cards correctly.

If I had more time to work on this task, I would try to increase the speed of the card identification by doing more processing on the cut out cards before the identification method. For example, my final code just crops a square of fixed size around the the center of the card. This creates a much larger image than is required for the identification. By using the rectangle_finder and rotating the frame, I could crop and resize the cut out card to an amount that optimizes the speed and still produces accurate identification.

A pattern I have found with my card identifier is cards are far more likely to be identified as a card that has many training images, such as card ID 33. Limiting the number of cards of the same ID from the training images would also speed up the identification as less comparisons would be required.