EEE6219 - Computer Vision – Model Solutions – 2014/15

Question 1

a) Explain the differences between face detection and face recognition. What conditions would make face recognition a challenging problem? [4 marks]

The objective of face detection is to localise any and all faces within an image. Face recognition seeks to discriminate between one face and another. Face detection therefore typically precedes face recognition.

A number of factors make face recognition the challenging problem: changes of viewpoint and illumination, facial expression, ageing of the subject, the subject wearing cosmetics or glasses, or growing a beard or moustache. All of these factors affect the appearance of the face and hence present a difficulty for recognition.

b) Subspace learning approaches are commonly used for face recognition. Briefly describe the two principal approaches employed. Which one would you expect to be better? Justify your answer. [4 marks]

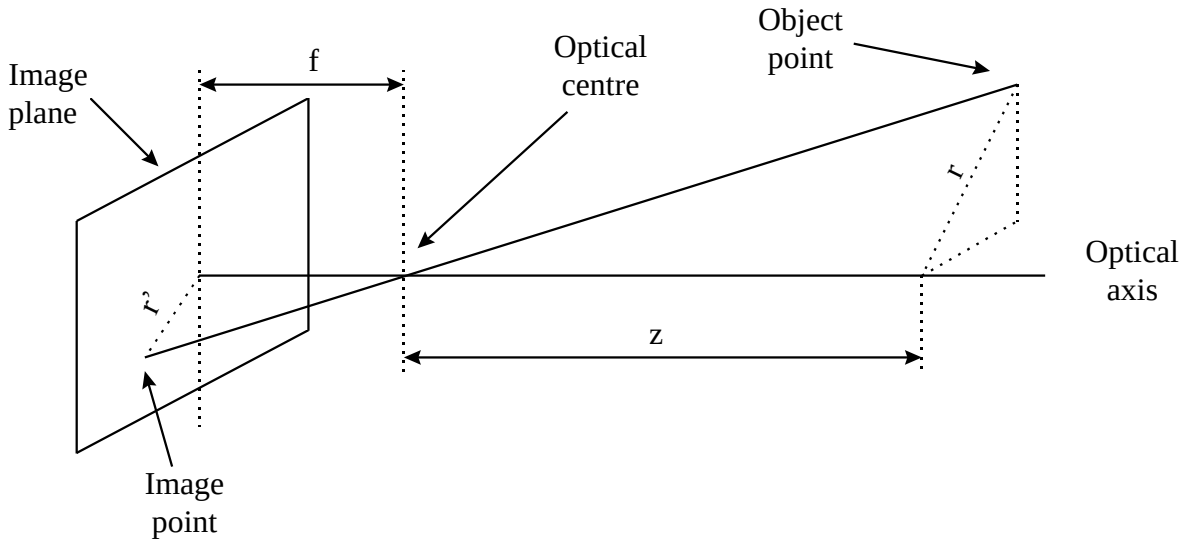
The two main subspace learning approaches commonly used in face recognition are: principal component analysis (PCA), and Fisher linear discriminant analysis (LDA) also sometimes known as Fisher faces.

PCA is an unsupervised approach which aims to identify the directions in multidimensional measurement space which have the largest variation; successive orthogonal directions of increasing variability are extracted. By discarding the components which have noise/small/insignificant variation, dimensionality reduction can be achieved.

LDA seeks to identify projection in the measurement space which maximises the scatter between classes but minimises the scatter within classes.

You should expect LDA to perform better because it is a supervised technique which uses the information from the class labels. PCA, on the other hand, is unsupervised. In fact, there is absolutely no reason to assume that the direction of largest variation obtained from PCA is in any way helpful for discriminating between different classes.

c) Using a suitable diagram, describe the key geometric features of the pinhole camera model. Derive a simple expression relating the projection of a point on the sensor plane the corresponding point in the object space. [4 marks]



By similar triangles: $\frac{r'}{f} = \frac{r}{z}$

$$\therefore r' = \frac{rf}{z}$$

What is the principal shortcoming of the pinhole camera? How does a real camera differ from this model and what are the main effect encounters a consequence of the non-pinhole model?

The principal shortcoming of the pinhole camera model is that the aperture is of infinitesimal area and so in practice would not transmit any light! Practical cameras include a lens of finite aperture but this means that not all light rays pass through the optical centre of the camera leading to a finite depth of field within which points in object space can be brought to a focus.

d) What is the objective of histogram equalisation of the grey levels and a monochrome image? Describe the steps involved in histogram equalisation. What this histogram equalisation do to the appearance of the image? [8 marks]

The objective of histogram equalisation is to modify the distribution of grey-levels in an image such that the fullest use is made of the dynamic range of intensity. Typically it would be applied to images where, due to deficiencies in the lighting/camera aperture setting, the grey-level values were all grouped at one end of the dynamic range.

The steps involved in histogram equalisation are as follows:

1. Form a histogram of the pixel intensities in the input image, *p(X)*.

2. For every value *x',* in the input histogram calculate the value of the summation:

$$\sum_{x_{\min}}^{x'} p(X) = K \text{ , say}$$

i.e. add up all the bin contents for $x \leq x'$.

3. Determine the corresponding value of *y'* from:

$$K = \frac{NM\left(y' - y_{\min}\right)}{y_{\max} - y_{\min}}$$

4. Replace all the pixels which have the value *x'* in the input image with the value *y'* in the output image.

Histogram equalisation typically makes images which have regions where detail is not readily visible due to being too dark or too light become more acceptable to the human observer. The effect is similar to changing the lighting conditions under which the original image was acquired.

Question 2

a) Object recognition methods fall into two broad classes: global-based methods, and parts- or local-based methods. Explain the basic properties of these two approaches. Compare the advantages and disadvantages of both object recognition methods. [2 marks]

Global-based methods of object recognition attempts to construct a single constructor for the whole object. The recognition of the object is then achieved by a single match to an exemplar vector of features.

Parts- or local-based approaches seek to identify a set spatially-localised component features, forming a 'signature' for each local feature. Recognition of the whole object is achieved by identifying a suitable conjunction of the component features that, together, indicate the presence of the object.

Global approaches have the advantage of using all the available information to form an object signature. The component signatures extracted with parts-based methods, on the other hand, individually use a limited amount of information and may therefore be ambiguous in certain cases. In addition, the need, in parts-based recognition, to identify a given spatial organisation of components is an extra complicating step in the processing. The major disadvantage of global approaches is that they only work for unoccluded views of the object in question. If part of the object is hidden from view, the extracted signature vector could be very different from the unoccluded object hence making matching very difficult. Parts-based methods, however, are able to cope with partial occlusion of the object.

b) Describe the principal processing steps in the implementation of the Canny edge detector. What of the three distinct design objectives set out by Canny and how does each of the processing steps in the algorithm achieve these goals? [8 marks]

The Canny algorithm comprises the following steps:

1. The image is smoothed by convolving with a 2D Gaussian function (typically implemented as two 1D convolutions).

2. The x- and y-gradient components are calculated and the total gradient magnitude computed.

3. The gradient magnitude map is non-maximally suppressed *normal* to the edge directions. Often this is achieved by a crude quantisation of edgel directions and the application of a 3x3 mask.

4. From the non-maximally suppressed magnitude map, edges are labelled as such if they exceed an upper threshold value.

5. In recognition of the fact that some edges may not be labelled due to noise, candidate edgels which exceed a lower threshold value are identified and neighbouring sites examined using and edge following algorithm to see if there is supporting evidence for labelling them as edges. Namely, is the marginal edgel surrounded by clear edgel candidates? If so, the marginal edgel is labelled as an edge.

6. Edge strings shorter than some predefined length are deleted since they are probably caused by noise.

7. Edge locations are computed to sub-pixel resolution *normal* to the edge direction either by: fitting a polynomial to the gradient magnitude map (without non-maximal suppression) and identifying the turning point, or estimating the mean of some magnitude distribution normal to the edge.

Canny's design criteria were:

a) To maximise the edge localisation accuracy.

b) To maximise the signal-to-noise ratio.

c) To obtain of single pixel width.

(a) and (b) are approximately achieved by the Gaussian filtering stage (1) due to the special properties of the Gaussian function.

(c) is achieved by the non-maximal filtering stage.

Explain why the Canny edge detector produces unpredictable results in the neighbourhood of corners. [2 marks]

The Canny detector can give rise to unpredictable results in the neighbourhood of corners since here the edge orientation is uncertain. Thus the non-maximal suppression stage which is carried-out normal to the edgel can produce unpredictable results due to the gross errors in estimating the normal direction.

c) In relation to characterising a detector, describe what you understand by the following terms:

i. True positives
ii. False positives
iii. True negatives
iv. False negatives

The terms: true positive, also positive, true negative, and false negative can best be explained in the following table:

	True state of nature		
Predicted state		True	False
	True	True positive	False positive
	False	False negative	True negative

Thus, for example, if the true state of nature as a positive and the detector predicts a positive, the detector's prediction is correct and this is a true positive. If, on the other hand, the true state of nature is false but the detector predicts true, the detector is in error and this is a false positive. If the true state of nature is false and the detector correctly predicts false, then this is a true negative. Finally, if the true state of nature is true that the detector predicts false, this is clearly in error and is a false negative.

How would you estimate these quantities? [4 marks]

To estimate these quantities requires a labelled data set. Each data record is presented to the detector and the detector's prediction recorded. Thus the fraction of true positives can be calculated from the quotient of the number of true data records correctly labelled by the detector, and the total number of true records in the labelled dataset. Calculation of the other quantities proceeds in an analogous manner.

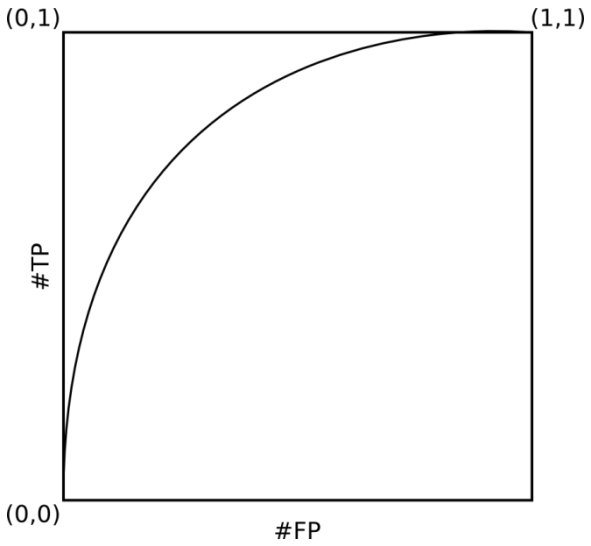
Also in relation to characterising a detector, what is a receiver operating characteristic (ROC)? [4 marks]

A receiver operating characteristic is a plot the fraction of true positives over the labelled dataset (ordinate) against the fraction of false positives (abscissa). In general, the ROC plot will join the points (0, 0) and (1, 1). For the case of the scoring classifier, this plot is obtained by varying the threshold of the true decision between zero and infinity.

Sketch typical ROC plots for a 'good' detector, and a detector that performed no better than random guessing. What features of the ROC plots suggest a 'good' detector?

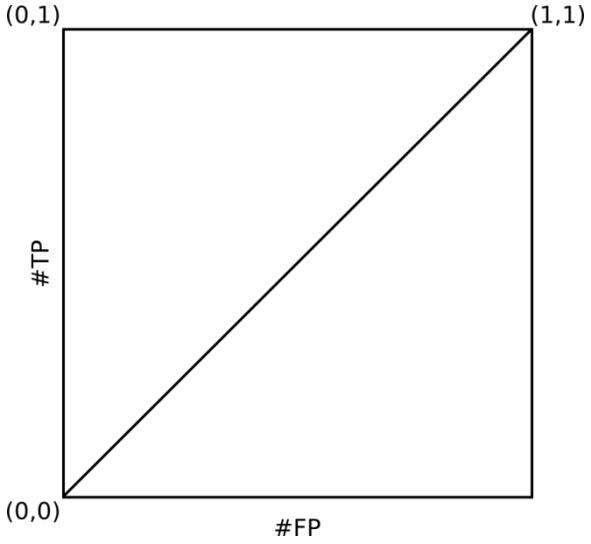
Ideally, the ROC plot should abruptly rise from the point (0, 0) to the point (0, 1), and thence horizontally the point (1, 1). That is, the detector should label all true examples that generate no force positives independent of threshold. In practice, the best that can be hoped for is an ROC plot that 'pushes' as far the top left-hand corner of the ROC plot.

A typical receiver operating characteristic for a 'good' detector shown below.



Starting from the point (0, 0) decreasing the threshold, true positive fraction rises rapidly with only a small increase in false positive fraction.

If the detector does no better than random guessing, the ROC plot is as shown below. In this case, for a given threshold, there is a 50% probability that the detector will predict the right class label (and therefore a 50% probability that it will be wrong).



Question 3

Explain how the Hough transform can be used to detect straight lines of the form:

$$y = mx + c$$

in an image. (Assume that a suitable edge detection process has been performed on the image.) [8 marks]

Applying the Hough transform to the straight line equation yields:

$$c = y - xm \qquad \text{...(1)}$$

where *c* and *m* are now regarded as the variables and *x* and *y* are constants derived from the edgel locations in the image. Taking the Hough accumulator as a discretised *(c,m)*-space, every edgel at (x_i, y_i) yields a constraint in the *(c,m)*-space in the form of eqn.(1). Making a line entry for every (constraining) edgel in the image will result in a peak at the *(c',m')* value which describes the line in the image (or *peaks* where there are more than one line.)

How can the computational burden of making entries into the Hough accumulator be reduced? [4 marks]

The above scheme is computationally burdensome because a large number of entries have to be made in the Hough accumulator, one for every image edgel. This can be reduced by examining the gradient direction of each edgel which dictates the slope, *m*, of the straight line of which that edgel can be part. Thus each edgel now only requires a single accumulator entry.

Is it possible to estimate the length of the line *directly* from the Hough transform accumulator? [4 marks]

When the Hough transform entries for the edgels which make-up a line are made into the into the accumulator, they add in the requisite accumulator cell to yield a peak; the height of this peak is thus equal to the number of edgels which comprise the line. Thus, *in principle*, the line length can be estimated directly from the Hough accumulator although line fragmentation will reduce the peak height but fortuitous clutter could increase the peak height. So only the *approximate* length can be estimated.

Explain how you would determine the width of the bins in a Hough transform accumulator. [4 marks]

Bin width should ideally be dictated by the error of the x and y positions of the edgels such that the entry from an edgel is maybe ~90% certain to be made in the correct Hough accumulator cell. In practice, certain trade-off is involved since making the bins to large will result in an imprecise estimate the line parameters, while making the bins to small will, due to noise, make it difficult to identify a clear peak in the Hough transform.

Question 4

a) In the field of linear filtering of images, what is meant by a *separable filter*? Assuming an $M \times M$ image and an $N \times N$ kernel, calculate the *approximate* saving in the number of arithmetic operations required to perform a filtering operation as a two-dimensional convolution compared to performing the same filtering operation in separable form. For the case of $M = 512$, determine the kernel size for which it is computationally advantageous to switch to the separable implementation. [5 marks]

A separable filter is a two-dimensional filter kernel, *f(x,y)* which can be written:

$$f(x,y) = g(x) \times h(y)$$

In other words, it can be decomposed into the product of two functions, each of which is a function of only *x* and *y*, respectively. Such a filter can be implemented as two, one-dimensional convolutions.

Assuming an $M \times M$ image and an $N \times N$ kernel, implementing the filter operation as two-dimensional convolution would require approximately M^2N^2 multiply-accumulate operations. Implementing as two one-dimensional convolutions in separable form would require approximately $2M^2N$ operations. For the case of a M = 512 the 'break-even' point can be calculated by equating the two expressions:

$$M^2N^2 = 2M^2N$$

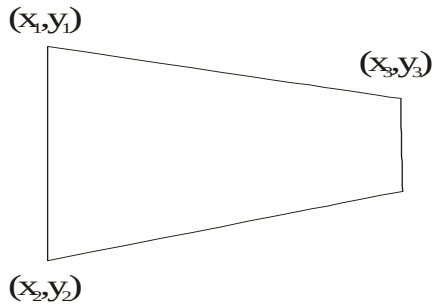
which has the solution of $N = 2$. Therefore, it is *always* preferable to implement a filter in separable form, if possible. (NB. The image size is a 'red herring'!)

b) It is desired to apply an affine transformation to the image in Figure 1 to produce a view approximately equivalent to that which would have been acquired with the optical axis of the camera normal to the front of the building. Explain how you could calculate a suitable affine transform based only on the information in the image. Carefully explain how you would determine the coefficients of the transform together with the effects of any assumptions you make. [9 marks]

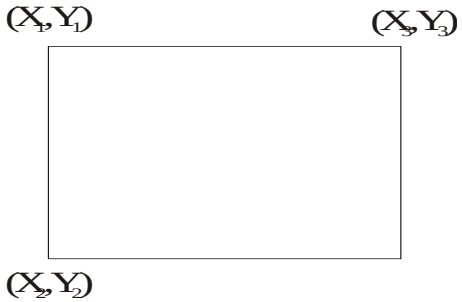
A suitable affine transform would take the form of:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

where a,b,c & d must to be estimated from the image alone. Using prior knowledge about the building, we can choose the pixel co-ordinates of the corners of three of the windows (since three points are sufficient):



and utilise the fact that in the *desired* view these window corners need to form a rectangle in the transformed image. In fact, it will be better used significantly more than three points and determine the coefficients using a least-squares procedure.



From this rectangle we can write down a number of constraints expressing the relationship between the rectangle's sides such as:

$$X_1 = X_2 \quad \& \quad Y_1 = Y_3$$

Explain any special precautions you would take in extracting relevant data from the original image in order to compute the transform. [3 marks]

The process of estimating transform parameters would involve the solution of a set of simultaneous equations, or equivalently the inversion of a matrix. In order not to render the matrix (near-) singular, you would need to take care that the selected data points were not collinear.

If you applied the transform calculated above to the image, what would the street lamp in front of the building look like in the transformed image? What does this tell you about the implied assumptions in an affine transformation? [3 marks]

The affine transform has been calculated from three points in a plane (*i.e.* on the surface of the building.) Thus all points in the image are assumed to be coplanar with the front of the building - any points which are not will suffer from some perspective distortion. In particular, the appearance of the street lamp under transformation will be behave as if it were painted on the front of the building and hence it will be transformed accordingly into a more elongated view of the lamp as in the following (approximately) transformed image.