Homework 4

Due: 27 Sep 2007

Problem 4.1

You are evaluating a piece of image recognition software that attempts to retrieve images of individual people from an image base and label them as male or female.

The evaluation corpus has 180 images in it. Of these, 60 are landscapes, 20 are crowded urbanscapes, 20 mugshots, 50 portraits, and 30 candid photos of individuals. The mugshots, portraits, and candids are each equally divided into pictures of men and pictures of women. The system yielded the following results:

	Guessed it was:	
Actually a:	male	female
Landscape	2	1
Urbanscape	2	3
Male Mugshot	8	2
Female Mugshot	3	7
Male Portrait	21	2
Female Portrait	3	21
Male Candid	6	1
Female Candid	0	8

a. With respect to the simple task "retrieve the individuals", what was the accuracy of the system, including both true positives and true negatives?

b. What was the system's precision and recall at the 'unlabelled' task of extracting images of individuals, without respect to whether it got the gender right?

c. What was the system's labelled precision and recall, i.e. taking into account the male/female label?

d. Think for a moment about what the system is trying to do here, and comment on its strengths and weaknesses. Keeping in mind that this is on a test corpus, what does the system's performance on various

subsets of the data tell you about its probable performance on real-world data?

As usual, show your work. Draw diagrams if you think it will help.

Problem 4.2

Consider an image with seven colours in the following pixel distribution:

transparent	9,000
opaque white	6,000
semitransparent white	600
light green	2,000
dark green	3,400
blue	1,000
black	3,000

This is to be run through a Huffman-based compression algorithm whose output file begins with the alphabet, with each colour having

- One byte containing the number of bits in a given colour's Huffman encoding
- Just enough bytes to contain that encoding, packed at the end with zeroes to the byte boundary
- Four bytes containing the raw red, green, blue, and alpha values for that colour

The end of alphabet is signalled with a null byte. After that null byte, the encoded file begins, using the Huffman code and ignoring byte boundaries.

a. Build a Huffman tree for these frequency counts. Draw it.

b. One section of one scan line in the file contains, in sequence, 2 transparent pixels, one semitransparent white pixel, two opaque white pixels, and five blue pixels. Assume it starts at a byte boundary, and encode this sequence in to the Huffman code implied by the tree you drew. (You can give the byte values either in raw binary or in hex; if in raw binary, at least mark the byte boundaries.)

c. If each uncompressed pixel takes up four bytes (one byte each for red, green, blue, and alpha), what compression ratio is being achieved in the sequence in the previous step? What is the compression ratio for the entire file?

d. What are the advantages and disadvantages of the given file format?

Problem 4.3 ($\times 2$)

On reflection, I do want you to complete last week's lab after all. For this problem, do Task 4 from Lab 3 (including resolving all three gotchas). If you aren't sure if you're correct up through Task 3, I'll be posting a solution to that around 4pm tomorrow (Tuesday).