**Course: Advanced Bioinformatics**

**Module title: Object Recognition**

**Module no. : 24**

**Object recognition** – Task (within computer vision) of finding and identifying objects in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different viewpoints, in many different sizes and scales or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems. Many approaches to the task have been implemented over multiple decades.

Classification deals with finding some particular information such as:

Does this contain cancer?

On the other hand detection is more related with the position of specific objects such as where are the cancer cells (if any)?

Finally, recognition or identification answers questions such as

Is this blue part is cancer cell?

Challenges of Object Recognition:

* View point variation
* Illumination

Similarly there are other challenges. Students are advised to search for examples of the following challenges from internet and textbooks of image processing.

* Occlusion
* Scale
* Deformation
* Modeling variability
* Within-class variations

Object Recognition Methods

There are two major categories for object recognition methods.

Appearance-based methods

- Use example images (called templates or exemplars) of the objects to perform recognition

- Objects look different under varying conditions:

* Changes in lighting or color
* Changes in viewing direction
* Changes in size / shape

- A single exemplar is unlikely to succeed reliably. However, it is impossible to represent all appearances of an object.

**1. Edge matching**

* Uses edge detection techniques, such as the Canny edge detection, to find edges.
* Changes in lighting and color usually don’t have much effect on image edges

Strategy:

1. Detect edges in template and image
2. Compare edges images to find the template
3. Must consider range of possible template positions

* Measurements:
* Good – count the number of overlapping edges. Not robust to changes in shape
* Better – count the number of template edge pixels with some distance of an edge in the search image
* Best – determine probability distribution of distance to nearest edge in search image (if template at correct position). Estimate likelihood of each template position generating image

**2. Divide-and-Conquer search**

* Strategy:
* Consider all positions as a set (a cell in the space of positions)
* Determine lower bound on score at best position in cell
* If bound is too large, prune cell
* If bound is not too large, divide cell into sub cells and try each sub cell recursively
* Process stops when cell is “small enough”
* Unlike multi-resolution search, this technique is guaranteed to find all matches that meet the criterion (assuming that the lower bound is accurate)
* Finding the Bound:
* To find the lower bound on the best score, look at score for the template position represented by the center of the cell
* Subtract maximum change from the “center” position for any other position in cell (occurs at cell corners)
* Complexities arise from determining bounds on distance

**3. Greyscale matching**

* Edges are (mostly) robust to illumination changes, however they throw away a lot of information
* Must compute pixel distance as a function of both pixel position and pixel intensity
* Can be applied to color also

**4. Gradient matching**

* Another way to be robust to illumination changes without throwing away as much information is to compare image gradients
* Matching is performed like matching greyscale images
* Simple alternative: Use (normalized) correlation

**5. Histograms of receptive field responses**

* Avoids explicit point correspondences
* Relations between different image points implicitly coded in the receptive field responses

**6. Large model bases**

* One approach to efficiently searching the database for a specific image to use eigenvectors of the templates (called eigenfaces)
* Model bases are a collection of geometric models of the objects that should be recognized

Feature-based methods

**1. Scale-invariant feature transform (SIFT)**

* Key points of objects are first extracted from a set of reference images and stored in a database
* An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors.

**2. Speeded Up Robust Features (SURF)**

* A robust image detector & descriptor
* The standard version is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT
* Based on sums of approximated 2D Haar wavelet responses and made efficient use of integral images.