6.1 Introduction

In the previous chapter, the focus was on simulating bottom-up mechanisms of visual attention. These define regions as interesting which have a high contrast to their surroundings and are unique in the setting. As mentioned in chapter 2, top-down influences also play an important role in human visual attention: knowledge, motivations, emotions, and goals define what is of interest in a certain situation. For example, drivers consider signboard in great detail rather than other parts of the road scene. While the influences of motivations and emotions are beyond the scope of this work, the topic of this chapter will be the goal-directed search for target region.

In human behaviour, bottom-up and top-down attention are always intertwined and may not be considered separately, although one may outweigh the other in certain situations. Even in a pure exploration mode, each person has own preferences resulting in individual scan-paths for the same scene. On the other hand, even if searching is highly concentrated for a target, the bottom-up pop-out effect is not suppressible. Despite its importance in the human visual system, top-down influences are rarely considered in computational attention systems. One of the reasons is that the neuro-biological foundations are not yet completely understood. Nevertheless, the extension of an attention system with top-down mechanisms is unavoidable if regions of interest shall be detected depending on a task. Moreover, the evaluation of the system is much easier with this extension since ground truth is available. An important model is VOCUS [32-37] that is able to regard top-down cues. In a learning phase, the system learns target-relevant features from a training image considering the properties of the target as well as the surrounding. In search mode, the system considers the information to excite or inhibit features and computes a target dependent top-down saliency map.
In this chapter, a novel detection system using visual attention model for the detection of signboard objects in real-world images has been incorporated. First, regions of interest are focused by an attention module, using either pure bottom-up attention (exploration) or the combination with goal-dependent top-down cues (visual search). Secondly, the region of interest is fed into a classifier detecting learned objects. In the proposed model a decision tree is used in learning phase to identify the most discriminant feature and those features are used to identify the target which in turn reduces the computational complexity and increases the efficiency of the system increases.

We briefly review the details of the model in section 6.2 with extensions in the same formal framework. In section 6.3 the performance of the system has been tested with various cases and in section 6.4 visual attention system with top down knowledge is compared with other attentions systems. In section 6.5 summary of this chapter is discussed.

6.2 System Description

Figure 6.1: Visual attention system with top down knowledge using decision tree classifier.
The block diagram in Figure 6.1 describes the flow of the system. It can be broadly divided into three major modules

1) The bottom-up module in the visual attention system identifies the most attended regions as explained in chapter 4 in the form of a saliency map

2) In top-down module of the feature extraction from the various maps generated by attention system for signboard detection are analyzed using decision tree classifier.

3) The feature selection module identifies the relevant features required for the signboard identification and uses for further classification. The following sections present the algorithm in detail

6.2.1 Visual Attention System

The input image is sub-sampled into a Gaussian pyramid on 6 different scales, and each pyramid level is decomposed into channels for red (R), green (G), blue (B), yellow (Y), intensity (I) using Equations (6.1) to (6.5). The orientation channels (O) are obtained from input image using oriented Gabor pyramids O(σ,θ), where σ∈[0..5] is the scale, and θ∈{0°, 45°, 90°, 135°} is the preferred orientation.

\[
I = \frac{r+b+g}{3} \tag{6.1}
\]

\[
R = r - \frac{g+b}{2} \tag{6.2}
\]

\[
G = g - \frac{r+b}{2} \tag{6.3}
\]

\[
B = b - \frac{r+g}{2} \tag{6.4}
\]

\[
Y = r + g - 2|r-g| + b \tag{6.5}
\]

Each feature is computed in a center-surround structure akin to visual receptive fields. Use of this biological paradigm renders the system sensitive to local
spatial contrast in a given feature rather than to amplitude in that channel. Center-Surround operations are implemented in the model as difference between a fine and a coarse scale for a given feature. The center corresponds to the pixel at the level $c \in \{2, 3\}$ in the pyramid. The surround corresponds to the pixel at the level $s = c+1$. Across scale difference between two maps, denoted as “$\Theta$”, is obtained by interpolation to the finer scale and point-by-point subtraction. The intensity feature type encodes for the modulus of image luminance contrast. That is the absolute value of the difference between the intensity at the center and the intensity in the surround as given in Equation (6.6)

$$F_{I,c,s} = N \begin{vmatrix} I(c) \big| \Theta \big| I(s) \end{vmatrix}$$  \hspace{1cm} (6.6)

The quantity corresponding to the double-opponency cells in primary visual context are then computed by center surround differences across the normalized color channels. Each of the three-red/green Feature map is created by first computing (red-green) at the center, then subtracting (green-red) from the surround and finally outputting the absolute value. Accordingly maps $F_{RG(c,s)}$ are created in the model to simultaneously account for red/green and green/red double opponency and $F_{BY(c,s)}$ for blue/yellow and yellow/blue double opponency and orientation using Equations (6.7) to (6.9)

$$F_{RG, c, s} = N \begin{vmatrix} R \ c \ - \ G \ c \ \Theta \ R \ s \ - \ G \ s \end{vmatrix}$$  \hspace{1cm} (6.7)

$$F_{BY, c, s} = N \begin{vmatrix} B \ c \ - \ Y \ c \ \Theta \ B \ s \ - \ Y \ s \end{vmatrix}$$  \hspace{1cm} (6.8)

$$F_{\Theta, c, s} = N \begin{vmatrix} O \ c, \Theta \ O(s, \theta) \end{vmatrix}$$  \hspace{1cm} (6.9)

The feature maps are then combined into three conspicuity maps, intensity $C_I$, color $C_c$, and orientation $C_o$ as shown in the Equation (6.10) to (6.12) at the saliency map’s scale ($\sigma = 4$). These maps are computed through across-scale addition ($\Theta$), where each map is reduced to scale four and added point-by-point:
To compute the orientation conspicuity map, four intermediary maps are created by combining the six feature maps. These intermediary maps are then combined into a single orientation conspicuity map.

\[
C_I = F_i \\
C_c = N\left(\sum_{i \in \mathcal{C}} F_i\right) \\
C_o = N\left(\sum_{i \in \mathcal{O}} F_i\right)
\]

(6.10) (6.11) (6.12)

To compute the orientation conspicuity map, four intermediary maps are created by combining the six feature maps. These intermediary maps are then combined into a single orientation conspicuity map.

\[
S = \frac{1}{3} \sum_{k \in \mathcal{C}, \mathcal{O}} C_k
\]

(6.13)

The \( N(.) \) represents the non-linear Normalization operator. The three conspicuity maps are normalized. The final Saliency map \( S \) is obtained using the Equation (6.13). From the saliency map the most attention regions are identified in the order of decreasing saliency based on the selective tuning model as discussed in Chapter 4.3.

6.2.2 Feature Extraction

The target in this case is a signboard and hence only the region where signboard is present is considered to be most salient region and is determined using bottom up visual attention system. From the Focus of Attention, 11 feature vectors are determined. The features are obtained from the values of the 7 feature maps \( (RG, BY, Intensity, Orientation 0^\circ, 45^\circ, 90^\circ, 135^\circ) \) and 3 conspicuity maps \( (Color, Intensity, Orientation) \) and 1 saliency map. It describes how much each feature contributes to the FOA. The classifier identifies whether the obtained feature vector from the target region matches with the actual target.

6.2.3 Feature Selection and Classification

To identify whether the salient region is the actual target, a decision tree classifier is used.
6.2.3.1 Decision Tree

A decision tree is a flow-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test and leaf nodes represent classes or class distributions. The topmost node in a tree is the root node. In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. A path is traced from the root to a leaf node which holds the class prediction for that sample. Decision trees can easily be converted to simple if-then rules.

6.2.3.2 Design of the Classifier

The algorithm uses an entropy-based measure known as information gain as a heuristic for selecting the attribute that will best separate the samples into individual classes. The fundamental structure is as follows:

1. The tree starts as a single node representing the training samples.

2. If the samples are all of the same class, then the node becomes a leaf and is labelled with that class.

3. Otherwise the algorithm uses an entropy-based measure known as information gain as a heuristic for selecting the attribute that will best separate the samples into individual classes.

4. A branch is created for each known value of the test attribute and the samples are portioned accordingly.

5. The algorithm uses the same process recursively to form a decision tree for the samples at each partition. Once an attribute has occurred at a node, it need not be considered in any of the node’s descendents i.e. an attribute is used only once.

6. The recursive partitioning stops only when any one of the following conditions is true:-
   
   (i) All samples for a given node belong to the same class
(ii) There are no remaining attributes on which the samples may be further partitioned. In this case, **majority voting** is employed. This involves converting the given node into a leaf and labelling it with the class in majority among samples.

(iii) There is no branch test-attribute i.e. all the attributes are used. In this case, a leaf is created with the majority class in samples.

The information gain is used to select the test attribute at each node in the tree. Such a measure is referred to as **attribute selection measure** or a **measure of the goodness of split**. The attribute with the highest information gain is chosen as the test attribute for the current node.

### 6.2.3.3 Expected Information

Let \( S \) be a set consisting of \( s \) data samples. Suppose the class attribute has \( m \) distinct values defining \( m \) distinct classes, \( C_i \) (for \( i = 1, \ldots, m \)). Let \( s_i \) be the number of samples of \( S \) in class \( C_i \). The expected information needed to classify a given sample is given by Equation (6.14).

\[
I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} (p_i \log_2 (p_i))
\]  

(6.14)

where \( p_i \) is the probability that an arbitrary sample belongs to class \( C_i \) and is estimated by \( s_i / s \).

### 6.2.3.4 Entropy

Let attribute \( A \) have \( v \) distinct values, \( \{a_1, a_2, \ldots, a_v\} \). Attribute \( A \) can be used to partition \( S \) into \( v \) subsets, \( \{S_{a_1}, S_{a_2}, \ldots, S_{a_v}\} \), where \( S_{a_j} \) contains those samples in \( S \) that have value \( a_j \) of \( A \). If \( A \) were selected as the test attribute (best attribute for splitting) then these subsets would correspond to the branches grown from the node containing the set \( S \). Let \( s_{ij} \) be the number of samples of class \( C_i \) in a subset \( S_{a_j} \). The entropy or expected information based on the partitioning into subsets by \( A \) is given by Equation (6.15)
\[ E(A) = \sum_{j=1}^{m} \frac{s_{1j} + \cdots + s_{mj}}{s} I(s_{1j}, s_{2j}, \ldots, s_{mj}) \quad (6.15) \]

where \( s \) is the total number of samples in \( S \), the smaller the entropy value is, the greater the purity of the subset partitions.

6.2.3.5 Gain

The encoding information that would be gained by branching on \( A \) is given by Equation 6.16.

\[ Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A) \quad (6.17) \]

The algorithm computes the information gain of each attribute. The attribute with the highest information gain is chosen as the test attribute for the given set \( S \). A node is created and the labelled with the attribute, branches are created for each value of the attribute and the samples are partitioned accordingly.

6.3 Decision Tree Classifier Algorithm

6.3.1 Training Phase

Features: - These are the 11 attributes based on which classification is done. They are obtained from the various stages of the visual attention system.

Class Label: - Here \( C_1 \) denotes the signboard class and \( C_2 \) denotes the other background information.

Training Dataset: - A set of tuples which contains feature values and class labels.

6.3.2 Process of Building the Decision Tree

The decision tree is built using the training dataset (Appendix D.4). The final decision tree (tree_table) representing the classifier adopts a matrix form where the positive entries denotes the column feature which is the immediate child of the current feature negative values denote the classes (leaf of the tree).
6.3.3 Testing Phase

In primary testing a set of test data is given whose class label is already known and the efficiency is calculated by checking if the output of the tree classifier matches the true class label. In secondary testing, the test sample is given to the classifier which will give the class label.

6.3.4 Pre-Processing Block

For the training phase images with sign board is considered. The image is split into sub windows as shown in Figure B.2. For each sub window the value of each feature is calculated. In this case eleven features are taken into consideration. They are:-

1. Color \((RG)\) Feature Map
2. Color \((BY)\) Feature Map
3. Color Conspicuity Map
4. Intensity Feature Map
5. Intensity Conspicuity Map
6. Orientation Feature Map in \(0^\circ\)
7. Orientation Feature Map in \(45^\circ\)
8. Orientation Feature Map in \(90^\circ\)
9. Orientation Feature Map in \(135^\circ\)
10. Orientation Conspicuity Map
11. Saliency Map

The values are then stored in tuple form. Each sub window represents a tuple. Class labels are then assigned to each tuple by the following criteria, if part of the sign board is present in the sub window then class 1 is given else class 2.
Figure 6.2: Image divided into sub windows with class labels 1 for signboard (yellow line) and 2 for non signboard information (red line).

6.3.5 Normalisation of the Sample Values

If the sample values are already normalized this block can be skipped else the following procedure is followed:

1) The number of branches for each attribute is decided by some heuristic method

2) The range of the dataset for each feature is then split up and normalized into the number of branches for that feature. For example, if a particular feature has dataset values 0.1, 0.9, 0.3, 0.8, 0.5, 0.1, 0.7 and the number of branches is 2 then the normalized form will be all values less than 0.5 is 1 and all values equal to or greater than 0.5 is 2, so the dataset becomes 1, 2, 1, 2, 2, 1, 2.
6.3.6 Steps in Building Decision Tree

1. Let \( t_{data} \) be the initial training sample set, \( n \) denote the number of features, \( n_{class} \) the number of classes.

2. Let \( feat \) array contain the number of branches for each feature.

3. Status array denotes whether the feature is used in the classifier or not, zero (0) denotes that the feature is not yet used in the building of the classifier and one (1) denotes the feature is used.

4. Find the Expected information of \( t_{data} \) using Equation 6.15.

5. Find the information gain for the features whose status is zero using Equation 6.16

6. The feature with maximal gain is chosen as the test attribute node. Make the status of that feature as one. Let the selected feature be \( f_{select} \)

7. Let \( num \) be the value at position \( f_{select} \) in the \( feat \) array. Split the dataset into \( num \) subsets such that each subset has the same values in the \( f_{select} \) column. For each subset do the following process:

   a. If the samples have the same class label \( C \) then let \( C \) be the class label for this leaf node. In \( tree_table \) matrix assign \( C \) to the branch value of parent attribute of the current attribute.

   b. If the samples have mixed class labels, depending on the conditions the steps are chosen

      i. If the number of tuples is less than a particular threshold or if all the features are used then let the leaf node contain the class with majority samples.

      ii. otherwise the following is done

         * Find the Expected information of subset data.
Find the maximum gain among the features which have status equal to zero. Let the selected feature be \( f_{\text{select}} \).

Make the status of the feature obtained in the previous step to one.

8. Step 7 is recursively performed. The process is stopped when any one of the following conditions are met:

i. No more tuples

ii. Status of all features is one.

### 6.3.7 Tracing the Decision Tree

For a given test data, tracing of the tree is done as follows:

a) Normalize the test sample tuple.

b) Let the root node be \( \text{tr}\_\text{feat} \).

c) Assign the value of the \( \text{tr}\_\text{feat} \) in the test tuple to \( \text{val} \).

d) In \( \text{tree}\_\text{table} \) matrix go to the “\( \text{tr}\_\text{feat} \)” column and the “\( \text{val} \)” row of the current attribute. If the value is negative then class label of test sample is the absolute of that value otherwise assign the value to \( \text{tr}\_\text{feat} \). Goto step c.

### 6.4 Experimentation and Evaluation

Figure 6.3.1 and 6.3.2 shows some sample of the signboards considered for testing the system. It includes signboards of pedestrian, crossing and bike [Appendix D.1 for the entire data set].
Figure 6.3.1: Sample images [a-b] for signboard identification system.
The analysis is done in two levels 1) Survey on signboard images using bottom-up visual attention system 2) Identification of discriminant features

6.4.1 Survey using Visual Attention System

First, a survey was done to identify whether the signboard is identified by the bottom-up visual attention system for the signboard dataset [17]. From figure 6.4, it is shown that the visual attention system identifies the signboard in some cases at the first level of iteration, in some at later levels and in few it failed. Table 6.1 gives a statistics of the results of bottom-up visual attention system.
Figure 6.4: Attention area identified by bottom-up visual attention system in various cases (a) first level (b) fourth level (c) failed.
Table 6.1: Results of bottom up visual attention system.

<table>
<thead>
<tr>
<th>Image type</th>
<th>Total number of images</th>
<th>Attention model</th>
<th>Failed to identify</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>First level</td>
<td>Other levels</td>
</tr>
<tr>
<td>Pedestrian</td>
<td>16</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Bike</td>
<td>16</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Crossing</td>
<td>16</td>
<td>2</td>
<td>13</td>
</tr>
</tbody>
</table>

6.4.2 Analysis on Feature Selection

Hence in this case the features obtained in the bottom up system are used as a training set for the classifier. To train the decision tree classifier, a set of eight samples of each type was considered. The bottom-up visual attention system was computed up to the saliency map.

Table 6.2: Training and testing samples for signboard detection.

<table>
<thead>
<tr>
<th>Image</th>
<th>Number of training samples</th>
<th>Number of testing samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedestrian</td>
<td>88</td>
<td>30</td>
</tr>
<tr>
<td>Crossing</td>
<td>75</td>
<td>20</td>
</tr>
<tr>
<td>Bike</td>
<td>175</td>
<td>40</td>
</tr>
<tr>
<td>Total</td>
<td>338</td>
<td>90</td>
</tr>
</tbody>
</table>

The eleven features from 7 feature maps (RG, BY, Intensity, Orientation 0°, 45°, 90°, 135°) and 3 Conspicuity maps (Color, Intensity, Orientation) and 1 saliency map were extracted and the class to which it belongs are taken as input data to train the classifier. The Table 6.2 shows the number of samples used for training and testing.
Figure 6.5: Decision tree classifier for signboard detection.
The Figure 6.5 shows decision tree classifier where the most discriminant feature to identify the class is shown. From the decision tree classifier, the hierarchy of the eleven features can be identified. The Table 6.3 shows the classification rate where, various combination of features were considered. The classification rate doesn’t show significant improvement when the number of features chosen were reduced. Hence the object signboard is able to achieve the same classification rate with two discriminant features namely color conspicuity map and the $90^\circ$ orientation feature map. There is no significant change in the classification rate by reducing the feature vector size from 11 to 2. Experiments show that the efficiency of the system is 98%. From Table 6.3, it is explicit that features of colour and orientation are sufficient to bring out a maximum efficiency for the application considered. The efficiency of the system doesn’t improve by increasing the number of features. Hence only two features are required to discriminant the road sign from the rest of the scene. The dataset which wasn’t classified properly were mostly in cases of, the background information being relatively dominant than the target information.

**Table 6.3: Classification rates for various feature selected.**

<table>
<thead>
<tr>
<th>Number of Features</th>
<th>Features Selected</th>
<th>Classification rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>3,8,7,2,4,6,9,1,5,10,11</td>
<td>98%</td>
</tr>
<tr>
<td>8</td>
<td>3,8,7,2,4,6,9,1</td>
<td>98%</td>
</tr>
<tr>
<td>7</td>
<td>3,8,7,2,4,6,9</td>
<td>98%</td>
</tr>
<tr>
<td>5</td>
<td>3,8,7,2,4</td>
<td>98%</td>
</tr>
<tr>
<td>4</td>
<td>3,8,7,2</td>
<td>98%</td>
</tr>
<tr>
<td>3</td>
<td>3,8,7</td>
<td>98%</td>
</tr>
<tr>
<td>2</td>
<td>3,8</td>
<td>98%</td>
</tr>
</tbody>
</table>

The knowledge that minimum two features namely color $RG$ Feature Map and Orientation $135^\circ$ FM are enough for recognizing the object signboard. The amount of contribution of the various features can be obtained from Table 6.4. This helps to identify the nature of inhibition and excitation levels required for these feature maps. This in turn helps in easily identifying the target (signboard).
6.4.3 Advantage of the Top-Down Visual Attention System using Decision Tree Classifier

The feature selection adopted here helps to reduce the search time involved in identifying the target object. Another added advantage to the system is that the classifier was able to detect signboard in cases where the bottom up visual attention system (saliency toolbox) failed as shown in Figure 6.6. The feature selection used before recognizing, reduces the search time as well give better recognition rate. The

Table 6.4: The hierarchy of features with its inhibition and excitation values to recognize the target

<table>
<thead>
<tr>
<th>No</th>
<th>Features</th>
<th>Level</th>
<th>Inhibition Value</th>
<th>Excitation Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Color RG FM</td>
<td>7</td>
<td>-</td>
<td>≤ 0.13</td>
</tr>
<tr>
<td>2</td>
<td>Color BY FM</td>
<td>6</td>
<td>≥ 0.9</td>
<td>&lt; 0.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt; 0.09</td>
</tr>
<tr>
<td>3</td>
<td>Color CM</td>
<td>1</td>
<td>= 0</td>
<td>0 - 53.32</td>
</tr>
<tr>
<td>4</td>
<td>Intensity FM</td>
<td>5</td>
<td>&lt; 0.38</td>
<td>≥ 0.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>≥ 0.38</td>
<td>&lt; 0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>≥ 0.54</td>
</tr>
<tr>
<td>5</td>
<td>Intensity CM</td>
<td>8</td>
<td>-</td>
<td>≤ 0</td>
</tr>
<tr>
<td>6</td>
<td>Orientation 0°FM</td>
<td>6</td>
<td>≥ 0.43</td>
<td>&lt; 0.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&lt; 0.89</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Orientation 45°FM</td>
<td>3</td>
<td>0.27 - 0.53</td>
<td>≥ 0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&lt; 0.27</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Orientation 90°FM</td>
<td>2</td>
<td>0.27 - 0.53</td>
<td>&lt; 0.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&gt; 0.53</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Orientation 135°FM</td>
<td>6</td>
<td>&lt;0.35</td>
<td>≥ 0.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt; 0.73</td>
</tr>
<tr>
<td>10</td>
<td>Orientation CM</td>
<td>9</td>
<td>-</td>
<td>≤ 0</td>
</tr>
<tr>
<td>11</td>
<td>Saliency Map</td>
<td>10</td>
<td>-</td>
<td>≤ 0</td>
</tr>
</tbody>
</table>
feature selection module followed by the decision tree classifier was able to identify the cases which were not identified due to the disadvantage in visual attention system. The classifier improves both time and quality of recognition. Hence the feature selection module used in top-down visual attention system enhances the systems performance.

6.5 Comparison with Other Attention Systems

6.5.1 Comparison with VOCUS

In VOCUS, the features for object recognition are as follows: intensity on/off, intensity off/on, orientation 0°, 45°, 90°, 135°, colors green, blue, red, yellow, Conspicuity intensity, orientation and color totally 13 features are used. In Figure 6.7 a test case image used by Simone Frintrop in the VOCUS model is shown with its feature values. It clearly highlights that all the 13 features used in the model don’t have very high significant values and only a few features like intensity and colour (red and yellow) have higher values compared to the others. This clearly indicates that the feature selection made in the model doesn’t identify the most discriminant feature necessary for the recognition task. So in the model discussed in this chapter, the decision tree identifies this discriminant feature among the rest of the feature and
uses for object recognition. Hence the feature vector size reduces to 2 compared to 13 with respect to VOCUS. The most important features are only used in this case which enhances the recognition rate and was also able to identify objects which was insignificant and was undetected in other models.

<table>
<thead>
<tr>
<th>Feature</th>
<th>weights (top)</th>
<th>weights (bottom)</th>
</tr>
</thead>
<tbody>
<tr>
<td>intensity on/off</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>intensity off/on</td>
<td>9.13</td>
<td>13.17</td>
</tr>
<tr>
<td>orientation 0°</td>
<td>20.64</td>
<td>29.84</td>
</tr>
<tr>
<td>orientation 45°</td>
<td>1.65</td>
<td>1.96</td>
</tr>
<tr>
<td>orientation 90°</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>orientation 135°</td>
<td>1.65</td>
<td>1.96</td>
</tr>
<tr>
<td>color green</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>color blue</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>color red</td>
<td><strong>47.60</strong></td>
<td>10.29</td>
</tr>
<tr>
<td>color yellow</td>
<td><strong>36.25</strong></td>
<td>9.43</td>
</tr>
<tr>
<td>conspicuity I</td>
<td>4.83</td>
<td>6.12</td>
</tr>
<tr>
<td>conspicuity O</td>
<td>7.90</td>
<td><strong>11.31</strong></td>
</tr>
<tr>
<td>conspicuity C</td>
<td><strong>17.06</strong></td>
<td>2.44</td>
</tr>
</tbody>
</table>

**Figure 6.7:** Extract from “Simone Frintrop’s test case” in [32].

Next, there is a difference in how training images are learned. In VOCUS, the algorithm is used to choose the training images which are suitable and enable a good representation of the target. In the proposed system, there is no specific algorithm for choosing training images, any random number of training images can be used.

In contrast, VOCUS learns the features after the scales have been summed up, yielding 13 feature values (2 intensity, 4 orientations, 4 colors, 3 Conspicuity maps). In contrast, the proposed system learns the features based on the scale determined from the saliency map and from the Conspicuity maps, yielding 11 feature values (1 intensity, 4 orientations, 2 colors, 3 Conspicuity maps, 1 saliency map).

VOCUS claims the scale of a target is not useful since in search mode the target should be detected at arbitrary scales, but is not investigated further. In the proposed system, the scale is based on the construction of the saliency map scale;
hence preference is given based on the target image, whereas VOCUS computes the most salient region within the object region and learns merely the features in this region.

VOCUS and the proposed system both regard the background for training the images.

In search mode, VOCUS computes an excitation and an inhibition map separately before joining them in the top-down saliency map. Furthermore, the top-down saliency map is separated from the bottom-up saliency map and the influence of each map is adjustable. In the proposed visual attention system, a feature selection analysis is performed based on the decision tree and only most contributing features are used in these exciting and inhibiting cues. This reduces the computational over head which is seen in the VOCUS.

6.5.2 Comparison with NVT

The system with the most similar approach to the top-down visual attention system by selective tuning model with top down mechanism extension is compared with NVT. The differences fall into three categories, firstly the differences concerning the bottom-up system, secondly the differences concerning the top-down mechanisms and thirdly the choice of the training image.

First, there are several differences in the bottom-up system. The computation of the feature maps are through parallel channels unlike NVT which is serial. There is usage of Selective tuning model for identification of the most salient region and for IOR not as WTA as in the case of NVT.

Secondly, there are differences in the top-down mechanisms in both the learning and search mode. In learning mode, NVT considers the whole region of the object, which is determined by the binary mask. The proposed system considers the saliency map of the image for learning. Another difference is the choice of features that are learned. NVT learns the features depending on the scale, i.e., it learns 42 feature values (red/green, blue/yellow, intensity and 4 orientations, each on 6 scales). In contrast, the proposed system learns the features based on the scale
determined by the saliency map and from the conspicuity maps, yielding 11 feature values (1 intensity, 4 orientations, 2 colours, 3 conspicuity maps, 1 saliency map).

Additionally, both systems consider target as well as background information for the learning of features but NVT considers 9 locations from a 3 x 3 grid of fixed size centred at the salient location whereas, the visual attention system with decision tree classifier regards the whole background. Probably, the approach of NVT is more biologically plausible since there is evidence that mainly the local neighbourhood of the target influences its salience. However, if the background has pop-out salient regions, NVT will fail to avoid those regions in the top-down model.

In NVT, weights the features directly based on exciting and inhibiting as well as bottom-up and top-down cues and then are mixed and directly fused into the resulting saliency map. In the proposed system, a feature selection analysis is performed based on the decision tree and only most contributing features are used rather than these exciting and inhibiting cues.

6.5.3 Comparison with Object Recognition System

In general, for all object recognition systems, the computation time involves searching the entire image space to identify the required target. Hence it can be modularized as two processes like segmentation and then recognition. Many other models consider visual attention system as a pre-processing step and then object recognition system as a front end which takes region of interest from the attention model and further compute features to identify the object. In visual attention system with decision tree, considers the entire task as a single job and orients the total system architecture to meet the object recognition task leading to a unique model.

In previous chapter the bottom-up visual attention system was compared with SIFT to identify the object; if $t_1$ is the time required to identify the salient region and $t_2$ is the time required to identify the object using SIFT.
Table 6.5: Comparison of computation time for various systems

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Visual attention system with decision tree classifier (ms)</th>
<th>Attention Model+ Object Recognition (ms)</th>
<th>Object Recognition System(SIFT) (Best Case) (ms)</th>
<th>Object Recognition System (SIFT) (Worst Case) (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>92</td>
<td>160</td>
<td>68</td>
<td>106</td>
</tr>
<tr>
<td>2</td>
<td>102</td>
<td>170</td>
<td>74</td>
<td>112</td>
</tr>
<tr>
<td>3</td>
<td>96</td>
<td>164</td>
<td>69</td>
<td>109</td>
</tr>
<tr>
<td>4</td>
<td>98</td>
<td>166</td>
<td>70</td>
<td>110</td>
</tr>
<tr>
<td>5</td>
<td>95</td>
<td>163</td>
<td>68</td>
<td>108</td>
</tr>
</tbody>
</table>

Figure 6.8: Graph depicting the time computation for various systems discussed in Table 6.5.

In case of top-down visual attention system, there is no $t_2$ time only $t_1$ time is required. Table 6.5 shows a comparative analysis of various models of object recognition system. The three combination considered are the visual attention...
system with decision tree, visual attention system as a back end with object recognition model as front end and a pure object recognition system. The comparative analysis clearly shows that minimum computation in case of attention system with decision tree. Hence the computation time for object recognition application can be drastically reduced from a quadratic time complexity to a linear time complexity if this combination model of top down knowledge with attention system is considered as shown in Figure 6.8.

6.6 Summary

The chapter discuss a novel approach for object detection using attention system with decision tree for feature selection and classification. The model when compared to VOCUS and NVT attention system improves in terms of performance and drastically decreases the computational complexity existing in the system. The model has been tested on a vehicle tracking application and the results are encouraging [3].

Remarks

The architecture discussed here has been published under the title, "Feature selection in top down Attention model" is published in Journal of Computing, ISSN 2151-9617, Vol.3, Issue 6, pp. 11-17, 2011.