Project

Vincent BODIN & Thomas MOREAU

Chosen topic. We decided to implement the paper by A. Mishra, K. Alahari and C.V. Jawahar called *'Top-Down and Bottom-up Cues for Scene Text Recognition'*[1] which deals with text recognition on typical street images. This work is done in parallel with the course *Object recognition and computer vision*. For this project we will focus on the graphical model part of the paper - model of the language.

Plan of the work. This project will focus on the part 3. of the paper *Recognizing Words*. We assume that the first part is done - it will be done for the project of *Object recognition and computer vision* - and we want to build a language model to be able to recognize words. We thus assumed we were able to extract windows that contain possible characters. Each window is associated to a goodness score (GS) as defined in the paper, note that a window could indeed be a true positive or a false one. In order to be able to classify correctly, we add a void label, say ϵ such that the set of class is $\mathcal{K}_\epsilon^n = \mathcal{K} \cup \{\epsilon\}$ - where n is the number of detection windows that potentially contain a character and $\mathcal{K} = \{c_1, \cdots, c_K\}$ is the set of possible characters (62 in English). The idea is to extract from those windows a way to re-construct the words that are originally written.

1. Word Model.
   - Graph construction: we build $G = (V, E)$ a graph. The procedure in the paper is the following: (a) order windows based on their horizontal locations; (b) add one node for every window sequentially from left to right (represented by X_i taking label x_i); (c) connect by edges nodes that sufficiently overlap.
   - Energy: for a possible word $\mathbf{x} = \{x_i | i = 1, \cdots, n\}$, we define an energy that is associated to this word as a sum of self and interaction energy:

     $$E(\mathbf{x}) = \sum_{i=1}^{n} E_i(x_i) + \sum_{\mathcal{E} \text{ edges}} E_{i,j}(x_i, x_j) \tag{1}$$

     The terms for energy are the following. If $x_i = c_j$ then the energy for this single character is $E_i(x_i = c_j) = 1 - p(c_j|x_i)$ where the likelihood $p(c_j|x_i)$ - which is also the confidence of the classifier - is learned by a SVM for instance. For the void label we set the energy being equal to:

     $$E_i(x_i = \epsilon) = \max_j p(c_j|x_i) \exp\left(-\frac{(\mu_{a_j} - a_i)^2}{2\sigma_{a_j}^2}\right) \tag{2}$$

     where a_i stands for the aspect ratio, μ_{a_j} the mean aspect ratio for c_j and $\sigma_{a_j}^2$ the variance of the aspect ratio for character c_j in the training data. As for the pairwise energy, it

[1] The paper can be downloaded at 'http://www.di.ens.fr/ alahari/papers/mishra12.pdf' and there is a web-page 'http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/'.

is defined through a prior lexicon that we will talk about afterward plus an overlapping interaction:

$$\begin{array}{rcl} E_{i,j}(x_i = c_i, x_j = c_j) & = & E^l(x_i, x_j) - \lambda_0 \exp\left(-\psi(x_i, x_j)\right) \quad (\forall c_i \neq \epsilon, c_j \neq \epsilon) \\ E_{i,j}(x_i = c_i, x_j = \epsilon) & = & \lambda_0 \exp\left(-\psi(x_i, x_j)\right) \\ E_{i,j}(\epsilon, \epsilon) & = & 0 \\ \psi_(x_i, x_j) & = & (100 - \mathrm{overlap}(x_i, x_j))^2 \end{array} \tag{3}$$

The final word is extracted by minimizing over all the possible energy. This is done in the paper by a sequential tree-reweighed message passing algorithm (TRW-S).

2. Computing Lexicon prior.

   - Bi-gram: it is a model where the lexicon prior, *i.e.* $E^l_{i,j}$ is learned from joint occurrences of characters is the lexicon. Denote by $p(c_i, c_j)$ the probability of the pair (c_i, c_j) then we simply set:
   $$E^l(x_i = c_i, x_j = c_j) = \lambda_l(1 - p(c_i, c_j)) \tag{4}$$
   where λ_l is a penalty for a character pair occurring.
   - Node-specific pair: the bi-gram model is not sufficient enough because it does not take into account the location for a pair to occur in a word. This is why a second langage model is used in this paper. It consist of cutting each lexicon word into m parts where m has to be defined (not explained in the paper). We treat then separately each 1/m part of the word to learn the pairwise cost. Now if we have a dictionary, say $\{\texttt{hello}, \texttt{hell}\}$ then the cost for `he` will be low if it occurs at the beginning of a word but high if it occurs in the other parts and the same thing applies to `ll` in second position. The underlying idea is to find regions of interest ($mathcalROI$). Formally the energy takes the following form:
   $$E^l(x_i = c_i, x_j = c_j) = \left\{ \begin{array}{ll} 0 & \text{if } (c_i, c_j) \in \mathcal{ROI} \\ \lambda_l & \text{otherwise} \end{array} \right. \tag{5}$$