Probabilistic Model of Object Detection Based on Convolutional Neural Network

Fang-Qi Li, Xu-Die Ren, and Hao-Nan Guo

School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
{solour_lfq}@sjtu.edu.cn

Abstract. The combination of a CNN detector and a search framework forms the basis for local object/pattern detection. To handle the waste of regional information and the defective compromise between efficiency and accuracy, this paper proposes a probabilistic model with a powerful search framework. By mapping an image into a probabilistic distribution of objects, this new model gives more informative outputs with less computation. The setting and analytic traits are elaborated in this paper, followed by a series of experiments carried out on FDDB, which show that the proposed model is sound, efficient and analytic.

Key words: Probabilistic Model, CNN, Object Detection.

1 Introduction

Detecting and locating local objects from images has been a challenging task in computer vision. Given some analytic, geometric patterns, this problem is handled by a branch of variants of Hough transform[1]. But detecting and locating of abstract objects, e.g.faces, have been analytically intractable.

So far, one of the most promising approaches is to combine convolutional neural network(CNN) with traditional methods. CNN has been taken as the most powerful mechanism for image processing, but since it was designed to solve classification problems firstly, its inflexibility in giving more informative outputs gave rise to redundancy and inefficiency of system structure in this task. The trade-off between efficiency and accuracy can be witnessed from the previous methods too. Sticking to the CNN structure as a classifier, those models served as deterministic models, which implies outputs that are less analytic.

The rest of the paper is organized as follows: Section 2 introduces traditional models' CNN structures and search frameworks. Section 3 introduces the structure and manipulation of a new probabilistic model. Experiments are presented and analyzed in section 4. Section 5 concludes the paper.

2 Related Works

2.1 Previous Models

As CNN outperforms other models in the task of recognizing abstract objects. A CNN combined with a search framework can provide a straightforward solution to this task: the search framework segments the whole image into pieces and feeds them into the CNN. However, applying CNN to every part of the image of every scale, known as exhaustive search[2] is computably expensive. An alternative approach, selective search[3] provides a heuristical segmentation of picture, which reduces complexity and accuracy. The duality of exhaustive and selective search depicts the compromise between accuracy and efficiency. As a variant of a dense artificial neural network, CNN's parameters are set to adapt to the innate trait of images, namely regional similarity, invariance under certain transformation, etc. Introducing of invariance by shared parameters increases the robustness and decreases the sensitivity at well. Thus an orthodox CNN is not well-designed to locate every trivial object from an image without a search framework.

The combination of CNN and a selective search framework shows extraordinary result[4]. The exactitude of locating by a cascade is discussed in[5]. A recent work provides an end-to-end framework under the assumption that different layers of CNN extract features of different scales[6]. However, the problems listed below shed light on a probabilistic model.

2.2 Problems

Traditional detection systems have two shortcomings. First, the process of vision recognization itself is not a deterministic process, but hitherto only deterministic models are proposed. They waste the regional traits of the targeted object and the consistency of images. Second, the search framework, a trade-off between efficiency and accuracy, lacks flexibility.

3 Probabilistic Model of Object Detection

3.1 Object Distribution

In order to use the incomplete information of a certain object, we propose a probabilistic model, together with an effective search framework and a heuristic option. As mentioned in [4], a deformation penalty can help to locate a local object better. The existence of a object within a scope of an image can be viewed as a probabilistic distribution(as the notion of electrons in quantum physics), i.e, the cars in Fig.1 can be seen as a distribution in Fig.2. Assuming the distribution is a normal distribution, then the existence of objects within an image can be illustrated by a mixture of Gaussian:

$$p(\mathbf{x}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \pi_n \frac{1}{2\pi} \frac{1}{|\Sigma_n|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \mu_n)^T \Sigma_n^{-1} (\mathbf{x} - \mu_n) \right\} \quad (1)$$

Where N, μ and Σ denote the number, locations and expansions, i.e.sizes of objects. We set $\pi = 1$ because this object distribution is not necessarily regularized. For a probabilistic model, instead of a regular classifier, a regression task is assigned to a CNN detector. In particular, the value of the target function is the density of probabilistic distribution at the center of the receptive field.

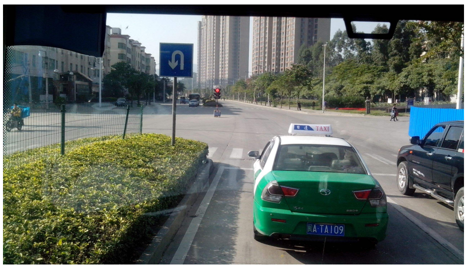


Fig. 1: Original Image

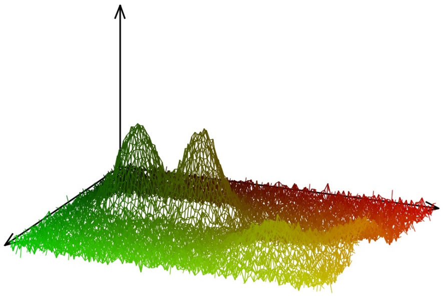
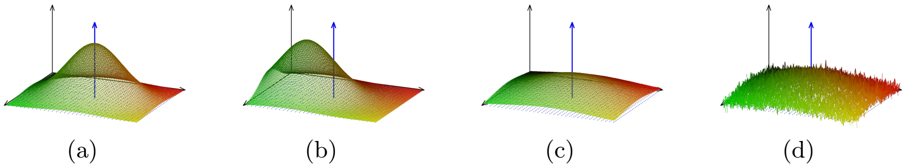


Fig. 2: Object Distribution of Cars

3.2 Reliability and Sensitivity of the CNN detector

We claim that such a CNN detector is only sensitive to the object of the appropriate size, i.e.an object with a suitable covariance matrix(Fig.3.(a)(b)). If an input contains too large an object, i.e.the covariance matrix has small eigenvalues or it zooms in on a part of an object, then the distribution becomes flat(Fig.3.(c)) and is likely to be submerged by noise(Fig.3.(d)). If an input contains too small an object, then the chance that it be placed in the middle of the receptive field which leads to a sharp output is relatively low. Compared with a classification detector, this detector gives outputs that are related to neighbor's, i.e.adjacent segmentations result in similar outputs. This enables the reduction of calculation, which is elaborated in section 3.4.



(a) (b) (c) (d)

Fig. 3: Output of the CNN Detector in Different Circumstances

3.3 Training of the CNN

Generated Dataset To train the regression function, we generate a dataset by deforming the targeted object and calculating the target values. By denoting the origin object as $\mathbf{x}$, the deformation as $s(s(\mathbf{x},0)=\mathbf{x})$, the target function as f, the deformation parameters as ϵ, we have a dataset generated by $\mathbf{x}$ as:

$$D_{\mathbf{x}} = \{(s(\mathbf{x},\epsilon), f(s(\mathbf{x},\epsilon)))\} \tag{2}$$

The error function of this CNN structure is thus given by:

$$E_0(\mathbf{w}) = \frac{1}{2}\sum_{\mathbf{x},\epsilon}\{y(s(\mathbf{x},\epsilon),\mathbf{w}) - f(s(\mathbf{x},\epsilon))\}^2 \tag{3}$$

Tangent Propagation It can be proved that training a model with a deformed dataset equals adding extra terms onto the error function, which is known as tangent propagation[7]. Tangent propagation was introduced to improve the robustness of traditional neural network against noise. In that sense, the error function is altered as:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1,\epsilon\to\mathbf{0}}^{N}\{y(\mathbf{x}_n,\mathbf{w}) - f(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\left\{\frac{\partial}{\partial\epsilon}y(s(\mathbf{x}_n,\epsilon),\mathbf{w}) - \frac{\partial}{\partial\epsilon}f(s(\mathbf{x}_n,\epsilon))\right\}^2 \tag{4}$$

Since:

$$\frac{\partial y}{\partial\epsilon} = \frac{\partial y}{\partial s}\frac{\partial s}{\partial\epsilon} \tag{5}$$

The derivatives of y with respect to x are elements of the Jacobian matrix, which can be traced by error propagation[8] and the derivatives of f are often analytically tractable, this method played an important role in image recognition before CNN came into fashion.

In the case of our task, an alteration can be made as an analogue. To simplify the task, we assume the variance matrix to be propotional to identical matrix:

$$\Sigma = \beta^{-1}I_2 \tag{6}$$

Taking the CNN's ability of handling rotation, we further assume that the shifting takes place in only one direction. Thus ϵ consists of ϵ_1 and ϵ_2 that denote the change of scale and location. As the derivates are found approximately around $\epsilon\to\mathbf{0}$, we can differentiate with respect to two parameters respectively.

First, a expansion in scale is denoted by $\beta \leftarrow \beta(1+\epsilon_1)$, thus the density at the center changes from $f(s(\mathbf{x},0))$ to:

$$f(s(\mathbf{x},\epsilon_1)) = \frac{1}{2\pi}\beta(1+\epsilon_1) = f(\mathbf{x})(1+\epsilon_1) \tag{7}$$

The derivative is inferred as:

$$\frac{\partial}{\partial\epsilon_1}f(s(\mathbf{x},\epsilon_1)) \approx \frac{f(s(\mathbf{x},\epsilon_1)) - f(s(\mathbf{x},0))}{\epsilon_1} = f(\mathbf{x}) \tag{8}$$

Seocnd, consider the change in location marked by ϵ_2:

$$f(s(\mathbf{x},\epsilon_2)) = \frac{1}{2\pi}\beta\exp\left\{-\frac{\epsilon_2^2}{2}\beta\right\} \tag{9}$$

Taylor expansion gives:

$$\frac{\partial}{\partial\epsilon_2}f(s(\mathbf{x},\epsilon_2)) \approx -\epsilon_2\beta f(\mathbf{x}) \tag{10}$$

It is necessary to take the second order of derivative that gives:

$$\frac{\partial^2}{\partial \epsilon_2^2} f(s(\mathbf{x}, \epsilon_2)) \approx -\beta f(\mathbf{x}) \tag{11}$$

The final error function takes the form:

$$\begin{aligned} E_f(\mathbf{w}) = E_0(\mathbf{w}) &+ \frac{\lambda_1}{2} \sum_{\mathbf{x}} \left\{ \frac{\partial}{\partial \epsilon_1} y(s(\mathbf{x}, \epsilon_1), \mathbf{w}) - f(\mathbf{x}) \right\}^2 \\ &+ \frac{\lambda_2}{2} \sum_{\mathbf{x}} \left\{ \frac{\partial^2}{\partial \epsilon_2^2} y(s(\mathbf{x}, \epsilon_2), \mathbf{w}) + \beta f(\mathbf{x}) \right\}^2 \end{aligned} \tag{12}$$

Theoretically, training with a generated dataset or a modified error function leads to the same effect. So either one can be chosen to train a CNN detector.

3.4 Search Framework

For a better compromise between efficiency and accuracy, we propose two frameworks of search that focus on accuracy and efficiency respectively.

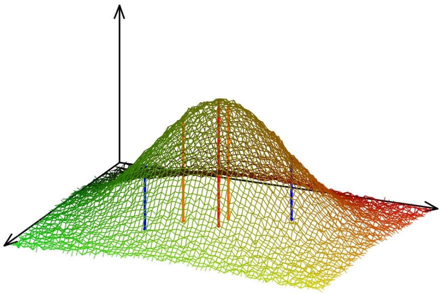


Fig. 4: Similirity from Adjancency

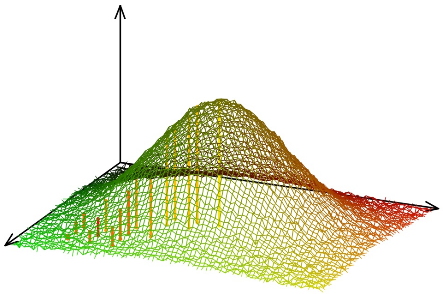


Fig. 5: An Ideal Search Direction

Exhaustive search Traditional exhaustive approach increases the size of the window by some pixels once, forming an arithmetic series, aiming at covering the continuous field of deformation with discrete steps of input resolution. The CNN detector based on regression task covers a quantified continuous range in deformation, namely the variance of the covariance matrix. By denoting the sensitive range of covariance for a CNN detector by $\beta^{-1}I_2 \sim (1+\alpha)^{-1}\beta^{-1}I_2$(this is equivalent to setting threshohld in Fig.4), for two discrete sizes of input s_n and $s_{n+1}(s_n \leq s_{n+1})$, it would be prefered if the CNN detector merges the variance seamlessly:

$$s_{n+1} \leq s_n(1+\alpha) \tag{13}$$

In another word, the sizes of segmentations can be chosen as terms from a geometric series, i.e.$s_0, s_0(1+\alpha), s_0(1+\alpha)^2, ...$, which is in accordance with intuition. Larger segmentations require a larger increase in pixels. Hence the calculation is reduced as the totality of segmentations is reduced from $O(N^3)$ to $O(N^2 \log N)$, where N denotes the width or height of the whole image.

Heurstical Search As the object distribution on the image is a scalar field and the CNN detector evaluates the scale of this field on a given point, it is possible to use the gradient information to reach the critical point. A heuristic automaton will detect the scale around itself randomly and runs in the direction with the best reward.(In Fig.5, the weakening of color depicts the process of searching). In dynamic images as videos, the automaton is capable of tracing the trajectory of a certain object.

4 Experiments

4.1 Training and Analyses

The FDDB dataset[9] is a labeled dataset with images containing one or more faces, the benchmark on FDDB requires recognition of local objects on a whole image, i.e.the aimed task. To generate the applicable set, we segment out 2041 faces, by shrinking/expanding(4082), shifting along axes(4000), to corners(4000), we generated a training set with 14123 images of 32 * 32. 12000 images are used to train the detector and the rest form the test set. We use a more straightforward demonstration by classifying non-confusing entries into one class, while the least square is used to train the CNN.

Three categories of CNN are introduced: C_1 consists of conv, pool, conv, pool, conv, dense; C_2 consists an extra pool before dense; C_3 is the same with C_1 in structure with an additional term in error function. C_1,C_2 and C_3 are denoted by red, green and blue in Fig.6 and Fig.7 for performace on the train set and test set after 500 times of iteration, statistics are showed in Table 1.

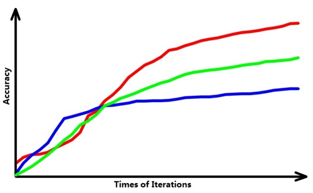


Fig. 6: Accuracy : 95.7%,84.9%,73.8%

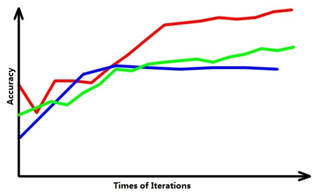


Fig. 7: Accuracy : 84.5%, 76.5%, 71.5%

An extra pooling layer reduces the accuracy as it causes extra loss of information. Introducing extra terms in error function increases the convergence speed at the initial phases, but it fails to react to sharp variances in the data set. But

Table 1: Performance of Models in Datasets w.r.t Iteration Times

Accuracy(%)	100 times		300 times		500 times	
	Train	Test	Train	Test	Train	Test
C_1	59.7	65.6	87.5	81.0	95.7	84.5
C_2	61.8	62.3	80.1	74.5	84.9	76.5
C_3	67.3	67.3	73.4	71.0	73.8	71.5

this method does not need a generated input (C_3 trained on 2041 original images uses the other 12082 images as test set, leads to the ordinary performance in both Fig.6 and Fig.7.)

4.2 Comparison

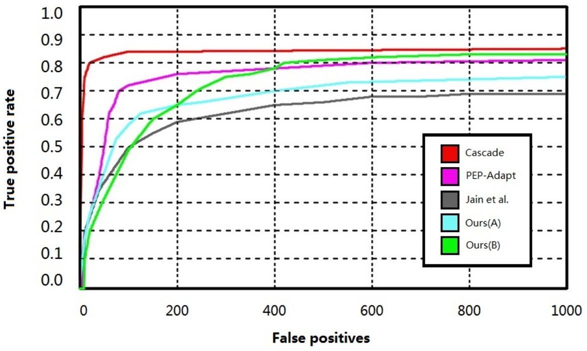


Fig. 8: Comparation on the FDDB dataset with state-of-the-art Models.

Using the discontinuous score on FDDB(where the size is taken according to the covariance matrix), we compare our models with state-of-the-art models including PEP-Adapt[10], Jain *et al.*[11] and the CNN Cascade. Model A uses naive exhaustive search framework. Model B is added with an additional evidence justifier, for each assumed object, calculations around it are done to see whether the object is mere noise. The results are showed in Fig.8.

Our model is second to only few models, e.g. the cascade for face detection. However, our model is much cheaper in computing complexity with only one CNN. Since the way we handle the detector's outputs is quite bald, it is possible that improvement in handling the output of the probabilistic detector, e.g.a Bayesian approach, can result in a better performance.

5 Conclusion

In this work, we propose a probabilistic model for object detecting. A CNN is assigned a regression task, which makes use of regional information for a more sophisticated output. It is elaborated that the deformation of objects, which has been an obstacle for object recognition, can become informative evidence for detecting and locating. This probabilistic model is open to more delicate processors, which makes better effects on this task attainable.

Acknowledgement

This research work is funded by the National Key Research and Development Project of China (2016YFB0801003), Key Laboratory for Shanghai Integrated Information Security Management Technology ResearchScience and Technology Project of State Grid Corporation of China (SGCC)

References

1. Duda R O, Hart P E. Use of the Hough transformation to detect lines and curves in pictures[J]. Ipsj Magazine, 15(1):11-15. (1972)
2. Kekre H B, Thepade S D, Parkar A. A Comparison of Kekre's Fast Search and Exhaustive Search for Various Grid Sizes Used for Colouring a Greyscale Image[C]// International Conference on Signal Acquisition and Processing. IEEE Computer Society:53-57. (2010)
3. Van d S K E A, Uijlings J R R, Gevers T, et al. Segmentation as selective search for object recognition[C] IEEE International Conference on Computer Vision. IEEE:1879-1886.(2011)
4. Ouyang W, Luo P, Zeng X, et al. DeepID-Net: multi-stage and deformable deep convolutional neural networks for object detection[J]. Eprint Arxiv, (2014)
5. Li H, Lin Z, Shen X, et al. A convolutional neural network cascade for face detection[J]. 5325-5334. (2015)
6. Cai Z, Fan Q, Feris R S, et al. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection[M] Computer Vision ECCV 2016. (2016)
7. Simard P, Lecun Y, Denker J S, et al. Transformation Invariance in Pattern Recognition Tangent Distance and Tangent Propagation[J]. International Journal of Imaging Systems & Technology 11(3):181197. (1998)
8. Bishop C M. Pattern Recognition and Machine Learning (Information Science and Statistics)[M]. Springer-Verlag New York, Inc. (2006)
9. Jain V, Learned-Miller E. FDDB: A Benchmark for Face Detection in Unconstrained Settings[M] UMass Amherst Technical Report. (2010)
10. Li H, Hua G, Lin Z, et al. Probabilistic Elastic Part Model for Unsupervised Face Detector Adaptation[C] IEEE International Conference on Computer Vision. IEEE:793-800. (2013)
11. Jain V, Learned-Miller E. Online domain adaptation of a pre-trained cascade of classifiers[C] IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society:577-584.(2011)