DeepDCF: The best feature for the DCF based tracker

QiangWang

November 22, 2016

1 Abstract

Discriminative Correlation Filters(DCF) have been widely used recent years in visual tracking due to the high precision and fast training and detection.

Convolutional Networks can enjoy millions labeled images for the offline representation learning.Classification even loss the spatial information.

Tracking is differenet with detection and classification. Classification and DET both eliminate the differences in the same class.

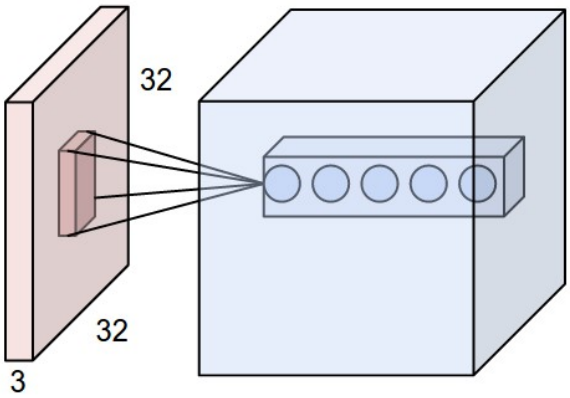
To put the tracking problem simple to the similarity metrics for total offline learning is a unreasonable hypothesis. The surrounding and the target motion fusing in temporal jointly determine the trajectory of the target.

2 Introduction

2.1 related work

2.2 CNN

CNN can be treated as a encode function $\Phi(x)$ of receptive field.



$$\Phi(x) = f^l$$

$x \in \mathbb{R}^{m \times n} \longrightarrow f \in \mathbb{R}^d$

It is sparse (only a few input units contribute to a given output unit) and reuses parameters (the same weights are applied to multiple locations in the input)

2.3 contributions

3 Discriminative Correlation Filters

3.1 DSST Derivation

(PCA-HOG+Gray)+DCF+Scale Esimation
key word: circular correlation,Parserval's identity, dense feature

3.1.1 Single Feature(gray)

patch: $x_1, ..., x_t$
label: $y_1, ..., y_t$
filter: w_t
test patch: z

$$\epsilon = \sum_{j=1}^{t} \|w_t \star x_j - y_j\|^2 = \sum_{j=1}^{t} \left\|\overline{W}_t \odot X_j - Y_j\right\|^2$$

$$W_t = \frac{\sum_{j=1}^{t} \overline{Y}_j \odot X_j}{\sum_{j=1}^{t} X_j \odot \overline{X_j}}$$

$$y = \mathfrak{F}^{-1}\left(\overline{W}_t \odot Z\right) = \mathfrak{F}^{-1}\left(\frac{\sum_{j=1}^{t} \overline{X}_j \odot Y_j \odot Z}{\sum_{j=1}^{t} \overline{X_j} \odot X_j}\right)$$

This is a little different between MOSSE[1] and this Derivation. No regularization term.

3.1.2 Multidimensional Features

patch: $f^l, l \in \{1, ..., d\}$ label: g filter: h^l test patch: z^l

$$\epsilon = \sum_{l=1}^{d} \left\| h^l \star f^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2$$

$$H^l = \frac{\overline{G} F^l}{\sum_{k=1}^{d} \overline{F^k} F^k + \lambda}$$

To obtain a robust approximation, here we update the numerator A_t^l and denominator B_t

$$A_t^l = (1-\eta) A_{t-1}^l + \eta \overline{G}_t F_t^l$$

$$B_t^l = (1-\eta) B_{t-1}^l + \eta \sum_{k=1}^{d} \overline{F}_t F_t^l$$

$$y = \mathfrak{F}^{-1}\left\{\frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B + \lambda}\right\}$$

3.2 CN

3.3 SRDCF

3.4 CF2

4 DeepDCF

The network in this filed utilize pooling to reduce the computation complexity and eliminate the spatial information. So we rebuild the network with

zero stride and total small size kernel convnet. We think the resolution in the spatial must be reserved. For the shortcomings of no stride and pooling, we have a small receptive field. So we anlynsis use the DilatedNet.

target patch: $x^l, l \in \{1, ..., d\}$ idea label: y filter: w^l test patch: z^l test output: g

For the learing part.

$$\epsilon = \sum_{l=1}^{d} \left\| w^l \star x^l - y \right\|^2 + \lambda \sum_{l=1}^{d} \left\| w^l \right\|^2$$

$$W^l = \frac{\overline{Y} \odot X^l}{\sum_{k=1}^{d} \overline{X^k} \odot X^k + \lambda}$$

$$g = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A^l} \odot Z^l}{B + \lambda} \right\}$$

$$g = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A^l} \odot Z^l}{B + \lambda} \right\} = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{\overline{Y} \odot X^l} \odot Z^l}{\sum_{k=1}^{d} \overline{X^k} \odot X^k + \lambda} \right\} = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} (Z^l \odot \overline{X^l}) \odot Y}{\sum_{k=1}^{d} \overline{X^k} \odot X^k + \lambda} \right\}$$

```
g=ifft2((sum(fft2(z).*conj(fft2(x)), 3).*fft2(y))./...
        (sum(fft2(x).*conj(fft2(x)), 3)+lambda));
```

The forward pass derivation should be familiar to us by now. There need some patience for the backward pass.

First, let's begining with some fundamental theorems of DFT and Complex-Valued Derivatives.

Because the DFT is a linear opreator, it's gradient is simply the transformation matrix it self. During the back-progation, then, the gradient is conjugated.[3]

$$Y = \mathfrak{F}\{y\}, \frac{\partial l}{\partial Y} = \mathfrak{F}\left\{ \frac{\partial l}{\partial y} \right\}$$

The second theorem we will use is Complex-Valued Derivatives[2]. Because we need pass δ throught the frequency domain. We will be more familar with the derivatives operator about complex .

$$\frac{\partial f(x, x^*)}{\partial x} = \overline{\frac{\partial f(x, x^*)}{\partial x^*}}$$

Now, Let's derive the formulas about the backward. For simplify, We start with $\frac{\partial l}{\partial g}$ and what we need is to results of $\frac{\partial l}{\partial x}$ and $\frac{\partial l}{\partial z}$.

$$G_{uv} = \frac{\sum_{l=1}^{d}(Z_{uv}^{l}\overline{X_{uv}^{l}})Y_{uv}}{\sum_{k=1}^{d}X_{uv}^{k}\overline{X_{uv}^{k}} + \lambda}$$

$$\frac{\partial l}{\partial Z_{uv}^{l}} = \frac{\partial l}{\partial G_{u,v}}\frac{\partial G_{u,v}}{\partial Z_{uv}} = \mathfrak{F}\left\{\frac{\partial l}{\partial g}\right\}_{uv}\frac{\overline{X_{uv}^{l}}Y_{uv}}{\sum_{k=1}^{d}X_{uv}^{k}\overline{X_{uv}^{k}} + \lambda}$$

$$\frac{\partial l}{\partial X_{uv}^{l}} = \frac{\partial l}{\partial G_{u,v}}\frac{\partial G_{u,v}}{\partial X_{uv}} = \mathfrak{F}\left\{\frac{\partial l}{\partial g}\right\}_{uv}\frac{\overline{Z_{uv}^{l}Y_{uv}}(\sum_{k=1}^{d}\overline{X_{uv}^{k}}X_{uv}^{k} + \lambda) - \overline{X_{uv}^{k}}(\sum_{l=1}^{d}(Z_{uv}^{l}\overline{X_{uv}^{l}})Y_{uv})}{(\sum_{k=1}^{d}\overline{X_{uv}^{k}}X_{uv}^{k} + \lambda)^2}$$

$$\frac{\partial l}{\partial x_{uv}^{l}} = \mathfrak{F}^{-1}\left\{\frac{\partial l}{\partial X}\right\}_{uv}^{l}$$

Algorithm 1 Forward pass: Calculate y

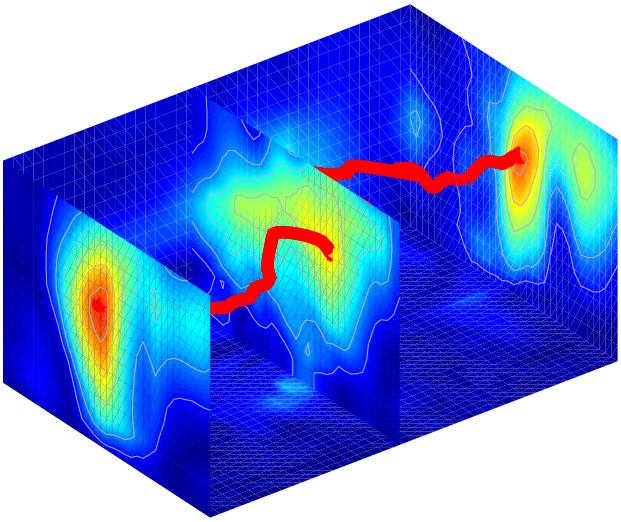
Require: target x, search z,initlize position p,ConvNet $\Phi(\cdot)$
  return y
  for $i := 1$ **to** n **do**
    crop and feature
    detection and update
  end for

Algorithm 2 Backward propagation: Calculate $\Phi(\cdot)$

Require: Gradient output $\frac{\partial l}{\partial y}$
  return Gradient input $\frac{\partial l}{\partial x}, \frac{\partial l}{\partial z}$

5 Experiments

I think that is a very perfect visualization of the training process.



5.1 Details and Paramemters

5.2 Baseline Comparison

Compare with handcraft feature. Compare different vgg(from shallow to deep) Compare different receptive field. Compare different Network(Alex,VGG,ResNet,tingNet) Compare different memory cost for the net struct.

5.3 OTB

In this benchmark,we should pay attention to raw results. Even We can training a very deep network like ResNet.

5.4 VOT

In this part, we should pay more attention in speed.

AR rank (EAO vs speed) should be presented.

6 Conclusions

We are the first to achieve end-to-end learning in DCF.

References

[1] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE, 2010.

[2] Are Hjørungnes. *Complex-valued matrix derivatives: with applications in signal processing and communications*. Cambridge University Press, 2011.

[3] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 2449–2457, 2015.