# Tracking

席茂  2019/01/12

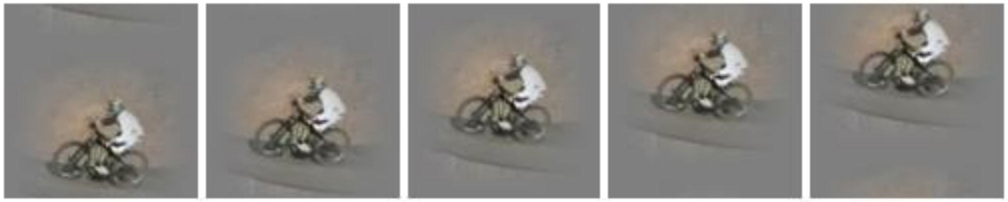
# Popular Methods

- Correlation Filter
- Siamese Networks
- Multi-Domain Networks
- Others(GAN, Reinforcement learning…)

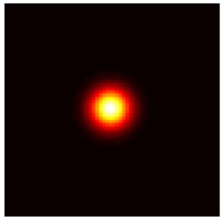
# Correlation Filter

## □ Kernelized Correlation Filters

basic idea: extract feature( hog, deep feature …) to solve a Ridge Regression with cyclic shifted samples



+30 +15 Base sample -15 -30



(a)

The complex closed form solution of Ridge Regression

$$\mathbf{w} = \left(X^H X + \lambda I\right)^{-1} X^H \mathbf{y},$$

Combine the cyclic matrix property the final solution is：

$$\hat{\mathbf{w}} = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}.$$

Next take the kernel trick,

# Correlation Filter

□ Kernelized Correlation Filters

Expressing the solution w as a linear combination of the samples：

$$\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i) \qquad \varphi^T(\mathbf{x})\varphi(\mathbf{x}') = \kappa(\mathbf{x}, \mathbf{x}')$$

The variables under optimization are thus α , instead of w

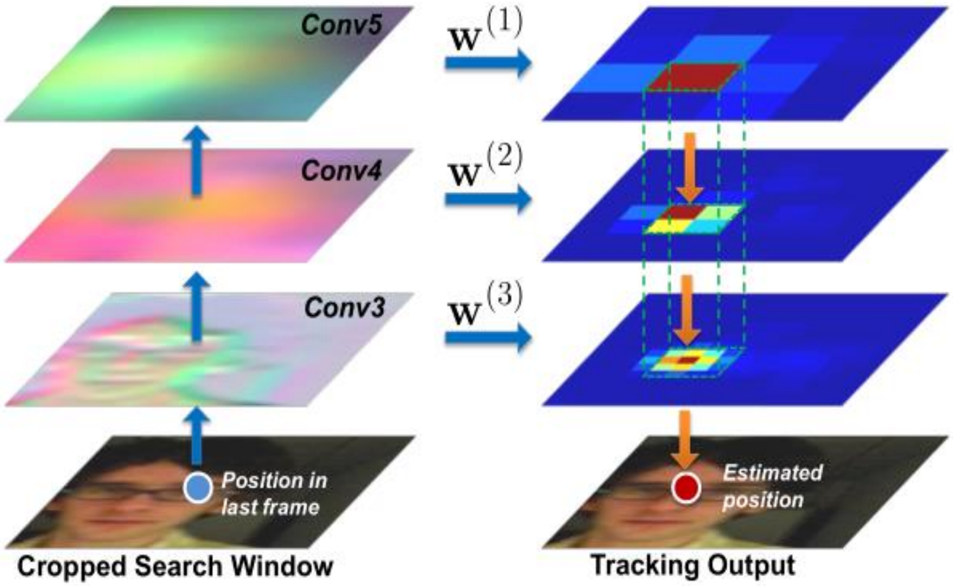
$$\boldsymbol{\alpha} = (K + \lambda I)^{-1} \mathbf{y},$$

Advantage： fast(100fps)， use different features

Disadvantage： can not handle scale variations

# Correlation Filter

□ Hierarchical Convolutional Features for Visual Tracking



通过结合不同层特征图生成多个响应图，最后通过coarse-to-fine找到目标响应最大位置

# Correlation Filter

同对提取特征图M × N × D（分别表示高，宽，与通道数），在M，N方向上进行循环移位获得样本，希望:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{m,n} \|\mathbf{w}\cdot\mathbf{x}_{m,n} - y(m,n)\|^2 + \lambda\|\mathbf{w}\|_2^2,$$

$$\mathbf{w}\cdot\mathbf{x}_{m,n} = \textstyle\sum_{d=1}^{D} \mathbf{w}_{m,n,d}^{\top}\mathbf{x}_{m,n,d}.$$

因此可得到滤波器频域的解:

$$\mathbf{W}^d = \frac{\mathbf{Y}\odot\bar{\mathbf{X}}^d}{\sum_{i=1}^{D}\mathbf{X}^i\odot\bar{\mathbf{X}}^i+\lambda}.$$

□ Coarse-to-Fine Translation Estimation

$$\underset{m,n}{\operatorname{argmax}} \quad f_{l-1}(m,n) + \gamma f_l(m,n),$$

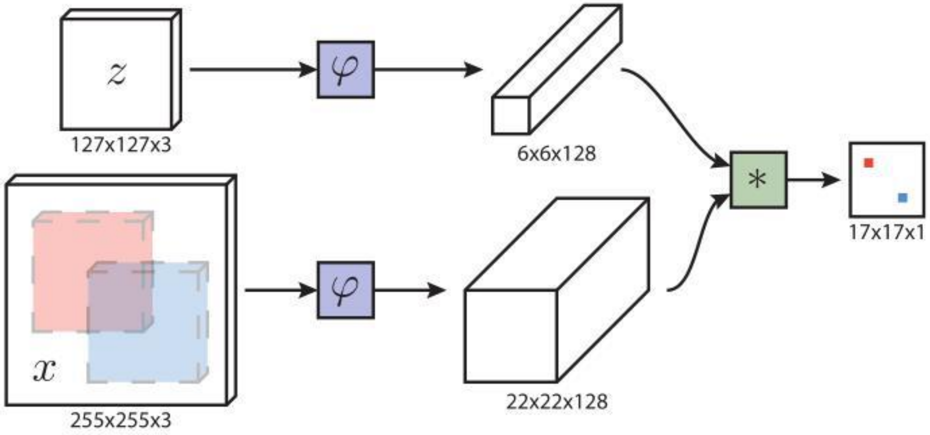
$$\text{s.t.} \quad |m-\hat{m}| + |n-\hat{n}| \leq r.$$

# Siamese Networks

□ Fully-Convolutional Siamese Networks（ECCV2016）

exemplar image z

search image x



The architecture is fully-convolutional with respect to the search image x. The output is a scalar-valued score map. This enables the similarity function to be computed for all translated sub-windows within the search image in one evaluation. In this example, the red and blue pixels in the score map contain the similarities for the corresponding sub-windows.

# Siamese Networks

□ Fully-Convolutional Siamese Networks（ECCV2016）

Loss：

$$\ell(y, v) = \log(1 + \exp(-yv))$$

where v is the real-valued score of a single exemplar-candidate pair and $y \in \{+1,-1\}$

$$L(y, v) = \frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{D}} \ell(y[u], v[u]) \ ,$$

requiring a true label $y[u] \in \{+1,-1\}$ for each position $u \in D$ in the score map

$$\arg\min_{\theta} \mathop{\mathbb{E}}_{(z,x,y)} L(y, f(z, x; \theta))$$

# Siamese Networks

□ Fully-Convolutional Siamese Networks（ECCV2016）

Tracking:

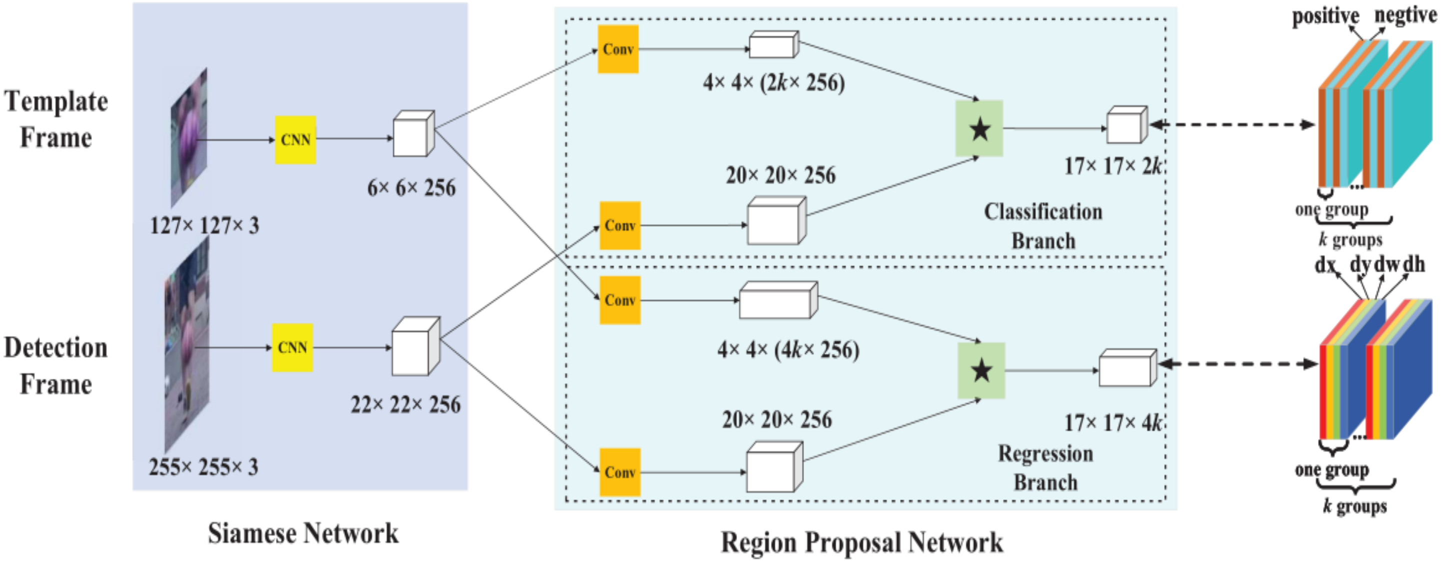
After get the score map, upsampling the score map using bicubic interpolation, from 17 × 17 to 272 × 272。To handle scale variations, we also search for the object over five scales $1.025^{\{-2,-1,0,1,2\}}$

Advantage：fast(100fps)，handle scale variations

Disadvantage：poor performance in average IoU and Location precise

# Siamese Networks

- □ SiameseRPN（CVPR2018）



每一组group代表一个不同宽高比的box

分类网络输出:每个位置$(x_i^{cls}, y_i^{cls}, :)$代表正或者负样本的label

回归网络输出:每个位置（$x_i^{\mathrm{reg}}, y_i^{reg}, :$)代表$dx, dy, dw, dh$,表示当前位置和groundtruth的距离

# Siamese Networks

□ SiameseRPN（CVPR2018）

Loss function

the classification branch adopts the cross-entropy loss

the regression branch：令 $A_x, A_y, A_w, A_h$代表预测的位置和形状， $T_x, T_y, T_w, T_h$代表groundtruth 。the normalized distance is:

$$\delta[0] = \frac{T_x - A_x}{A_w}, \quad \delta[1] = \frac{T_y - A_y}{A_h}$$

$$\delta[2] = ln\frac{T_w}{A_w}, \quad \delta[3] = ln\frac{T_h}{A_h}$$

经过平滑:

$$smooth_{L_1}(x, \sigma) = \begin{cases} 0.5\sigma^2 x^2, & |x| < \frac{1}{\sigma^2} \\ |x| - \frac{1}{2\sigma^2}, & |x| \geq \frac{1}{\sigma^2} \end{cases}$$

回归loss获得：

$$L_{reg} = \sum_{i=0}^{3} smooth_{L1}(\delta[i], \sigma)$$

# Siamese Networks

□ Loss function

最终优化：

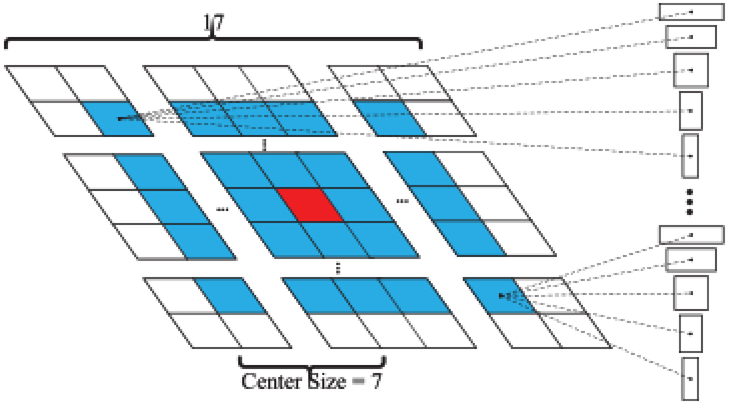
$$loss = L_{cls} + \lambda L_{reg}$$

□ Prediction

Collect top K points in all $A^{cls}_{w\times h\times 2k}$, so we can derive the corresponding anchor set as $ANC^* = \{(x_i^{an}, y_i^{an}, w_l^{an}, h_l^{an})\}$, next get K refinement coordinates $\{(\mathrm{x}_i^{reg}, y_i^{reg}, dx_l^{reg}, dy_l^{reg}, dw_l^{reg}, dh_l^{reg})\}$

□ Proposal selection

忽略距离中心点较远的候选区域，这里以中心点周围7*7的区域举例。



# Siamese Networks

## □ Results

results on VOT2015， VOT2016

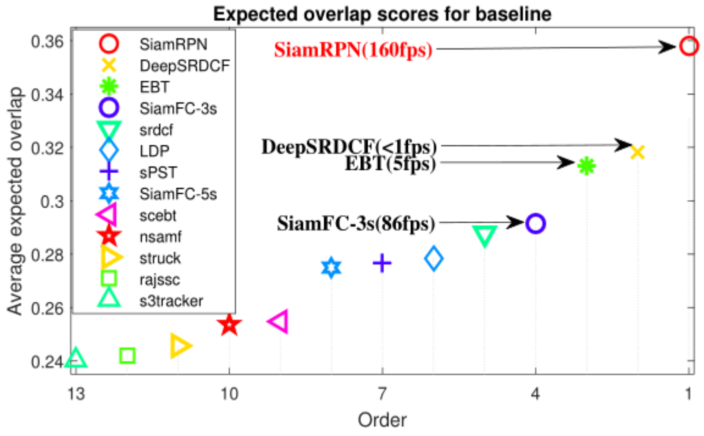
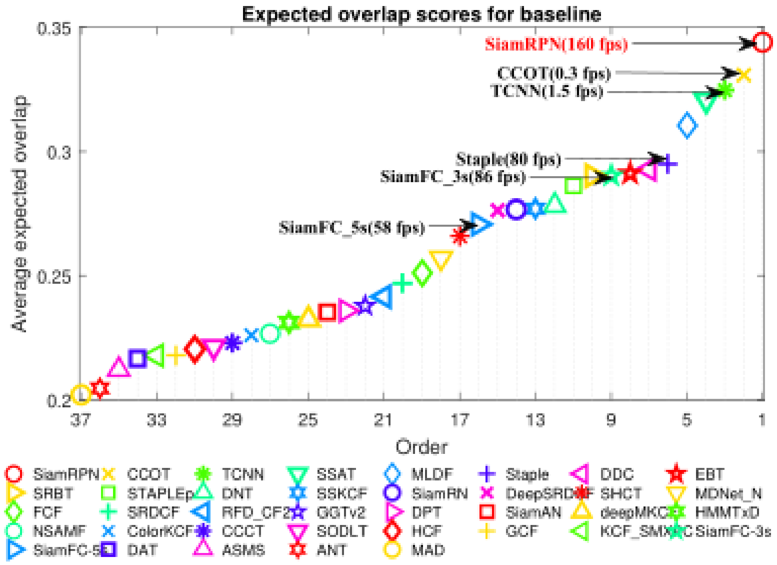


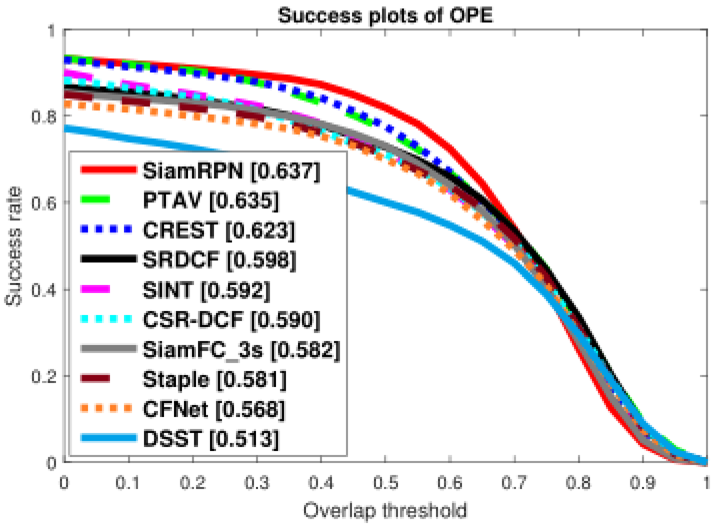
Figure 5: Expected overlap of our tracker, Siamese-FC and top 10 trackers in VOT2015 challenge.

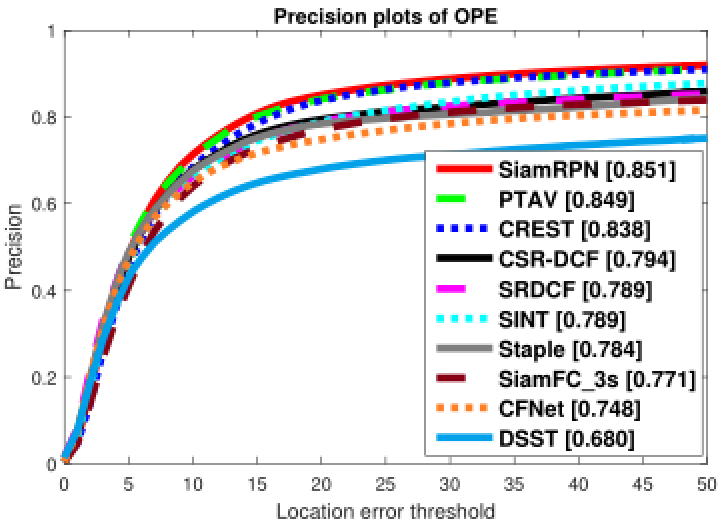


# Siamese Networks

□ Results

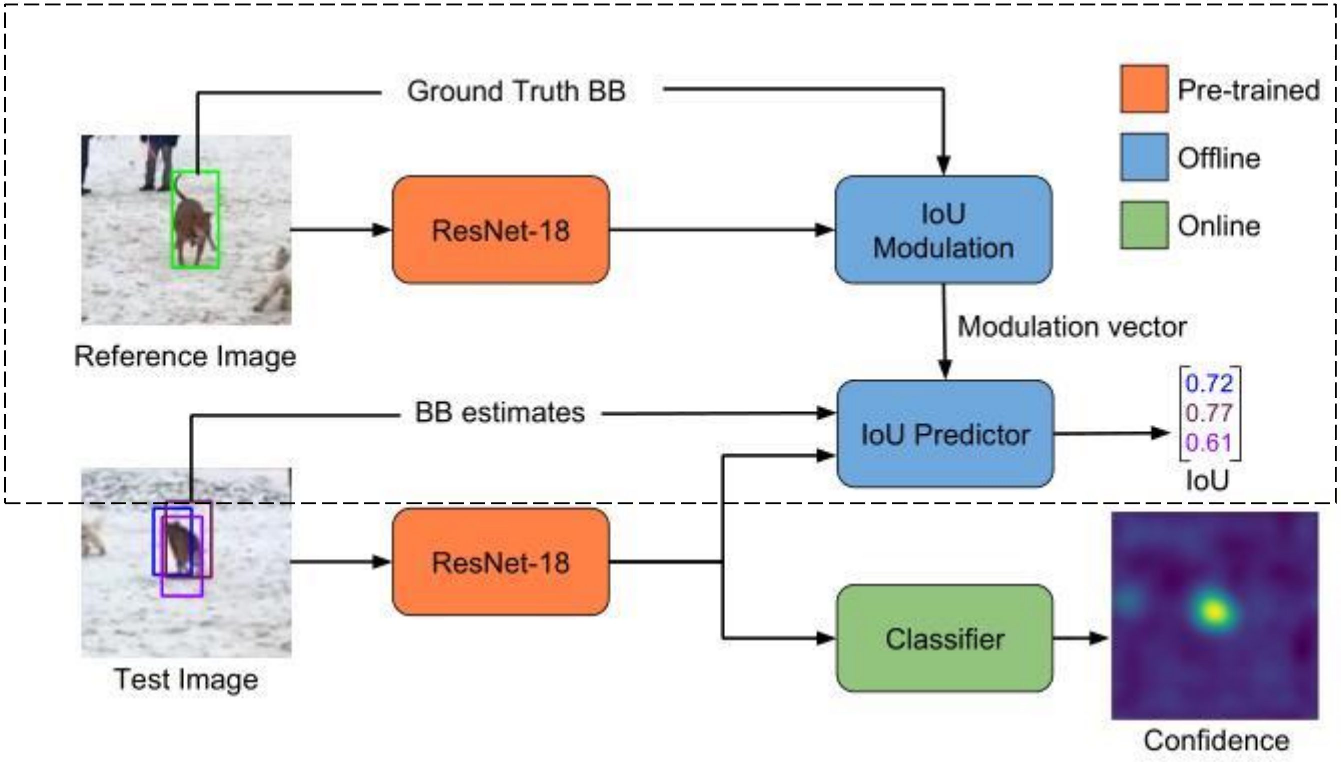
Results on OTB2015





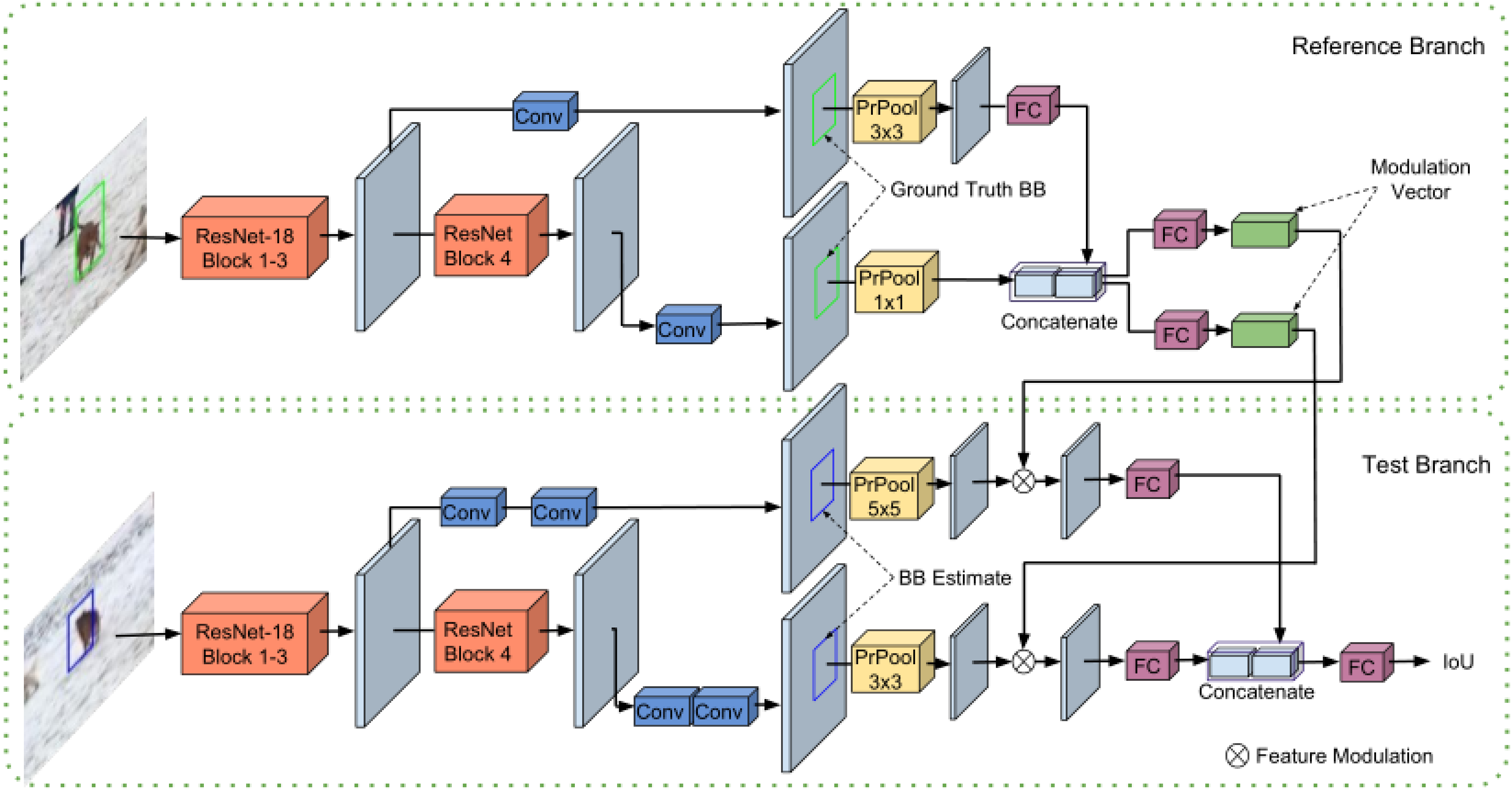
# Siamese Networks

□ ATOM : Accurate Tracking by Overlap Maximization(CVPR 2019)



主要是利用2018 detection中IoUnet的思想，通过最大化overlap以此达到性能提升。虚线内部分是预测目标尺度，虚线外部分预测目标位置

# Siamese Networks



# Siamese Networks

□ PrPooling（Precise RoI Pooling）

将离散的特征图插值为连续域特征图

$$f(x,y)=\sum_{i,j} IC(x,y,i,j)\times w_{i,j},$$

其中：$IC(x,y,i,j)=max(0,1-|x-i|)\times max(0,1-|y-j|)$

而$w_{i,j}$表示特征图中对应(i,j)的值。

池化过程:

$$\mathrm{PrPool}(bin,\mathcal{F})=\frac{\int_{y1}^{y2}\int_{x1}^{x2} f(x,y)\,dxdy}{(x_2-x_1)\times(y_2-y_1)}.$$

# Siamese Networks

□ Training

目标:最小化预测IoU与真实IoU的差值

使用LaSOT数据集选取样本图片对（两帧图片最多间隔100帧）

□ Target Classification by Fast Online Learning

分类模型为两层全卷积网络定义为:

$$f(x;w) = \phi_2(w_2 * \phi_1(w_1 * x)) \,.$$

Loss:

$$L(w) = \sum_{j=1}^{m} \gamma_j \| f(x_j;w) - y_j \|^2 + \sum_k \lambda_k \| w_k \|^2 \,.$$

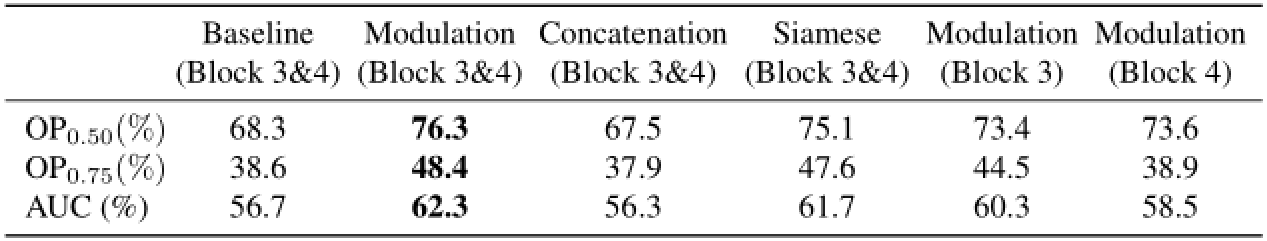
$y_j$由高斯函数生成

# Siamese Networks

□ Tracking

从前一帧预测的位置与尺度提取特征->利用分类模型得到目标的位置->结合前一帧目标的尺度加上随机噪声获取10个候选区->使用IoU prediction网络预测最大得分的三个候选框取平均

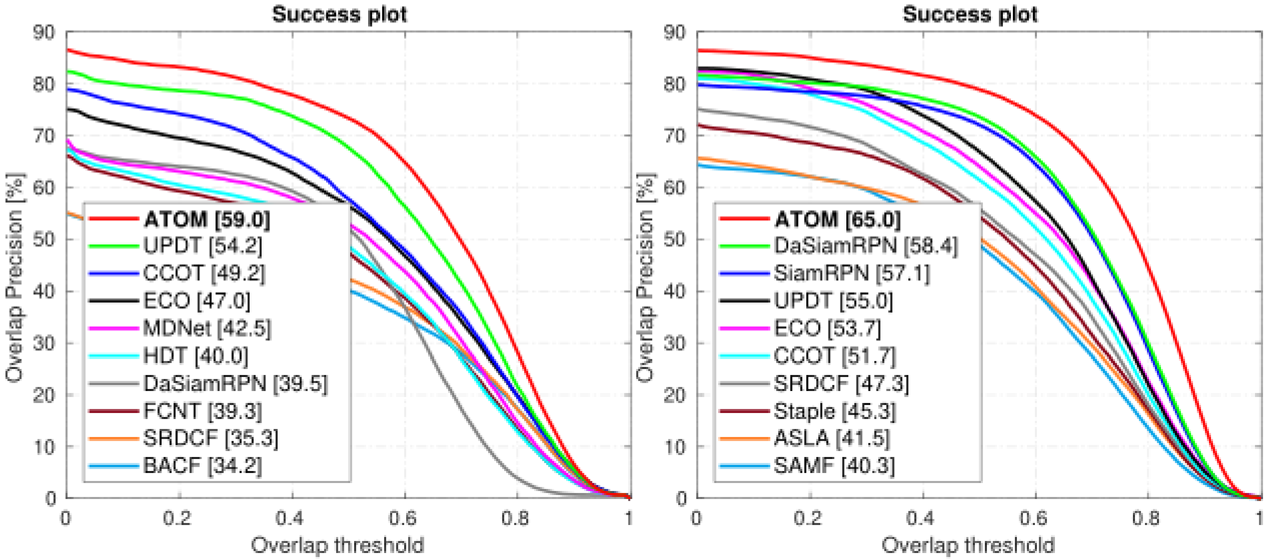
□ IoU Prediction Architecture Analysis



| | Baseline<br>(Block 3&4) | Modulation<br>(Block 3&4) | Concatenation<br>(Block 3&4) | Siamese<br>(Block 3&4) | Modulation<br>(Block 3) | Modulation<br>(Block 4) |
|---|---|---|---|---|---|---|
| OP$_{0.50}$(%) | 68.3 | **76.3** | 67.5 | 75.1 | 73.4 | 73.6 |
| OP$_{0.75}$(%) | 38.6 | **48.4** | 37.9 | 47.6 | 44.5 | 38.9 |
| AUC (%) | 56.7 | **62.3** | 56.3 | 61.7 | 60.3 | 58.5 |

# Siamese Networks

□ Results



(a) NFS

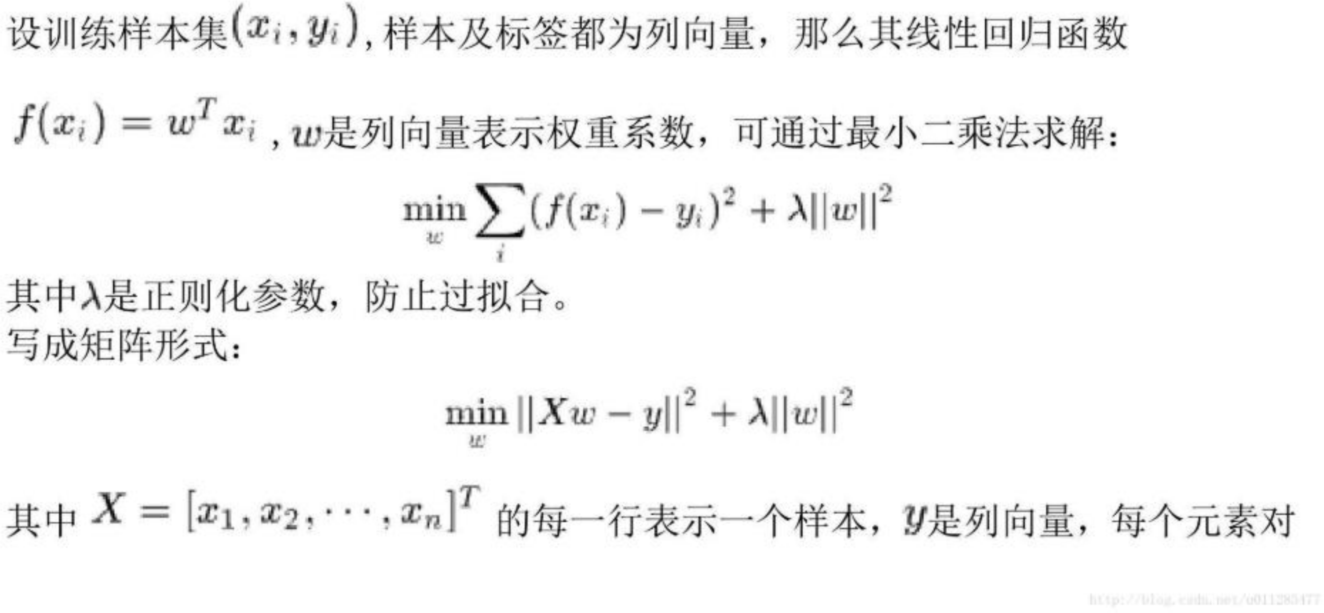
(b) UAV123

Figure 4. Success plots on NFS (a) and UAV123 (b). In both cases, our approach improves the state-of-the-art by a large margin.

# 论文汇总

- Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 583-596.
- Ma C, Huang J B, Yang X, et al. Hierarchical convolutional features for visual tracking[C]//Proceedings of the IEEE international conference on computer vision. 2015: 3074-3082.
- Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking[C]//European conference on computer vision. Springer, Cham, 2016: 850-865.
- Li B, Yan J, Wu W, et al. High Performance Visual Tracking With Siamese Region Proposal Network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8971-8980.
- Danelljan M, Bhat G, Khan F S, et al. ATOM: Accurate Tracking by Overlap Maximization[J]. arXiv preprint arXiv:1811.07628, 2018.
- Jiang B, Luo R, Mao J, et al. Acquisition of localization confidence for accurate object detection[C]//Proceedings of the European Conference on Computer Vision, Munich, Germany. 2018: 8-14.

# KCF推导



设训练样本集$(x_i, y_i)$, 样本及标签都为列向量，那么其线性回归函数

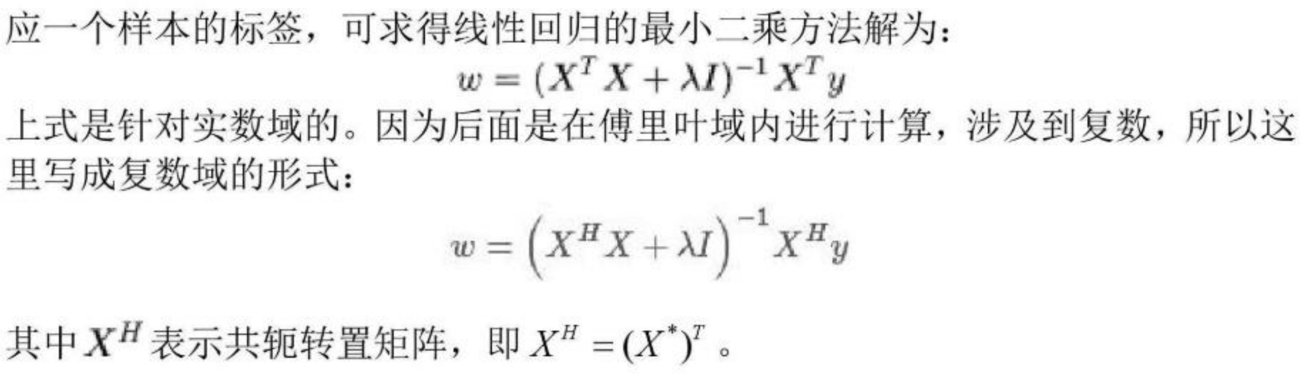
$f(x_i) = w^T x_i$ , $w$是列向量表示权重系数，可通过最小二乘法求解：

$$\min_w \sum_i (f(x_i) - y_i)^2 + \lambda ||w||^2$$

其中λ是正则化参数，防止过拟合。
写成矩阵形式：

$$\min_w ||Xw - y||^2 + \lambda ||w||^2$$

其中 $X = [x_1, x_2, \cdots, x_n]^T$ 的每一行表示一个样本，$y$是列向量，每个元素对



应一个样本的标签，可求得线性回归的最小二乘方法解为：

$$w = (X^T X + \lambda I)^{-1} X^T y$$

上式是针对实数域的。因为后面是在傅里叶域内进行计算，涉及到复数，所以这里写成复数域的形式：

$$w = \left(X^H X + \lambda I\right)^{-1} X^H y$$

其中$X^H$表示共轭转置矩阵，即 $X^H = (X^*)^T$ 。

# KCF推导

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

$$X = C(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}.$$

通过移位矩阵的性质，可通过下式得出矩阵X：

$$\{P^u\mathbf{x} \,|\, u = 0, \ldots n-1\}.$$

循环矩阵的性质告诉我们所有的循环矩阵可以由单行向量的傅里叶变换的对角元素表达，其中，$\hat{x}$表示x的傅里叶变换，如下：

$$X = F \operatorname{diag}(\hat{\mathbf{x}})\ F^H,$$

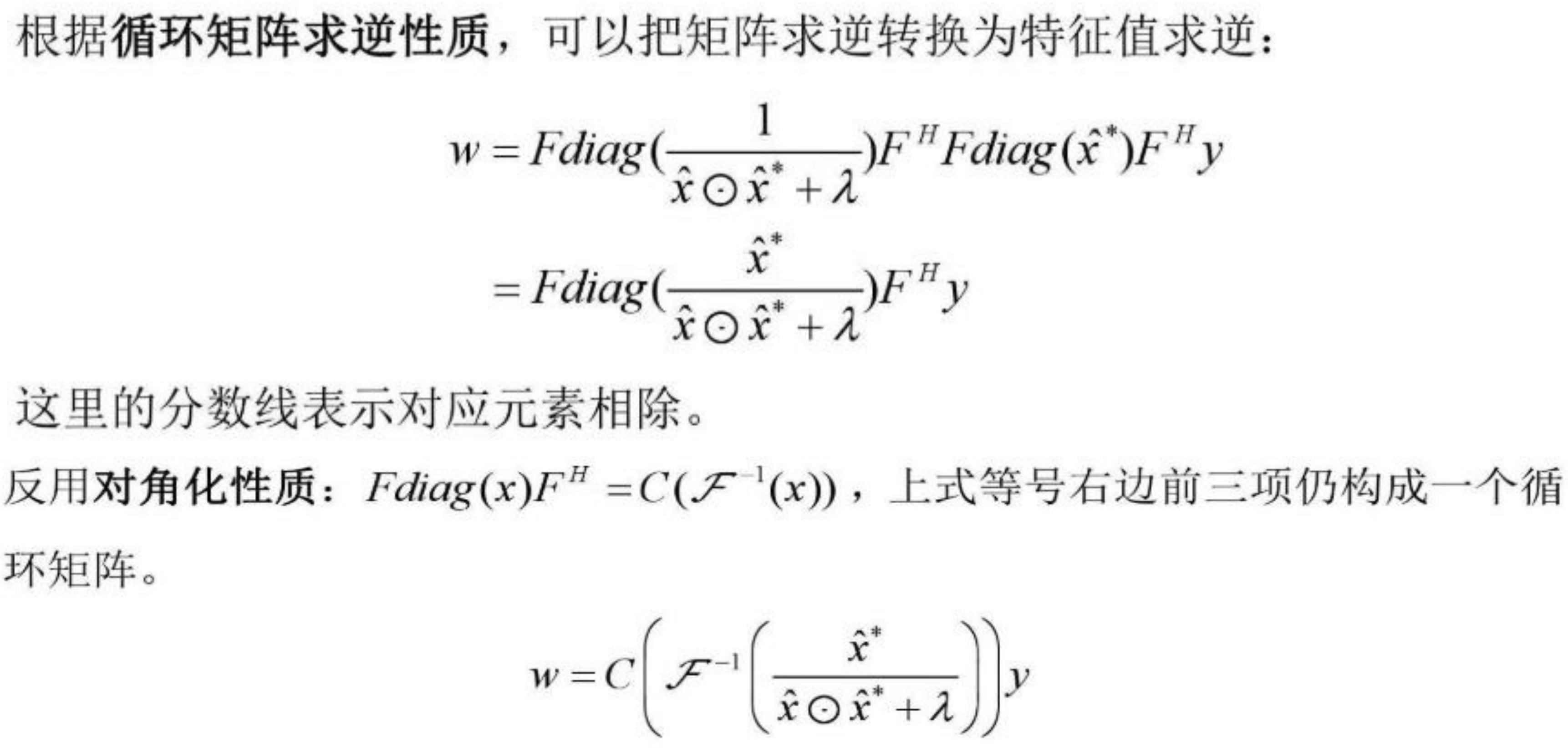
$$\mathcal{F}(\mathbf{z}) = \sqrt{n}F\mathbf{z}.$$

# KCF推导

将上述结果代入线性回归求解问题中得到:

$$\mathbf{w} = \left(F\mathrm{diag}\left(\hat{\mathbf{x}}^{*} \odot \hat{\mathbf{x}}\right) F^{H} + \lambda I\right)^{-1} X^{H}\mathbf{y}$$

其中⊙ 表示逐元素相乘



根据**循环矩阵求逆性质**，可以把矩阵求逆转换为特征值求逆：

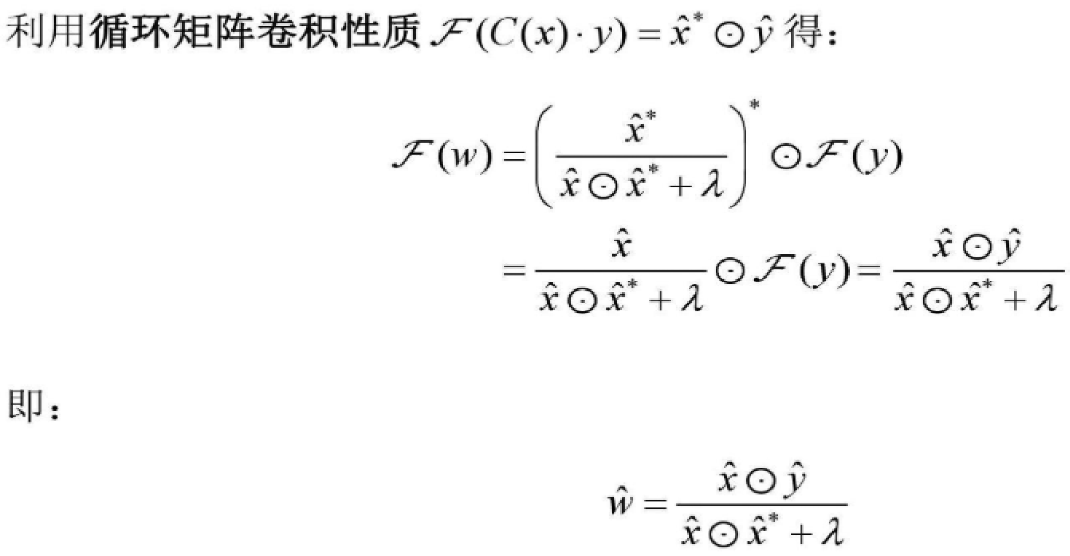
$$\begin{aligned} w &= Fdiag\left(\frac{1}{\hat{x}\odot\hat{x}^{*}+\lambda}\right)F^{H}Fdiag(\hat{x}^{*})F^{H}y \\ &= Fdiag\left(\frac{\hat{x}^{*}}{\hat{x}\odot\hat{x}^{*}+\lambda}\right)F^{H}y \end{aligned}$$

这里的分数线表示对应元素相除。

反用**对角化性质**：$Fdiag(x)F^{H} = C(\mathcal{F}^{-1}(x))$，上式等号右边前三项仍构成一个循环矩阵。

$$w = C\left(\mathcal{F}^{-1}\left(\frac{\hat{x}^{*}}{\hat{x}\odot\hat{x}^{*}+\lambda}\right)\right)y$$

# KCF推导



利用**循环矩阵卷积性质** $\mathcal{F}(C(x)\cdot y)=\hat{x}^{*}\odot\hat{y}$ 得：

$$\mathcal{F}(w)=\left(\frac{\hat{x}^{*}}{\hat{x}\odot\hat{x}^{*}+\lambda}\right)^{*}\odot\mathcal{F}(y)$$

$$=\frac{\hat{x}}{\hat{x}\odot\hat{x}^{*}+\lambda}\odot\mathcal{F}(y)=\frac{\hat{x}\odot\hat{y}}{\hat{x}\odot\hat{x}^{*}+\lambda}$$

即：

$$\hat{w}=\frac{\hat{x}\odot\hat{y}}{\hat{x}\odot\hat{x}^{*}+\lambda}$$