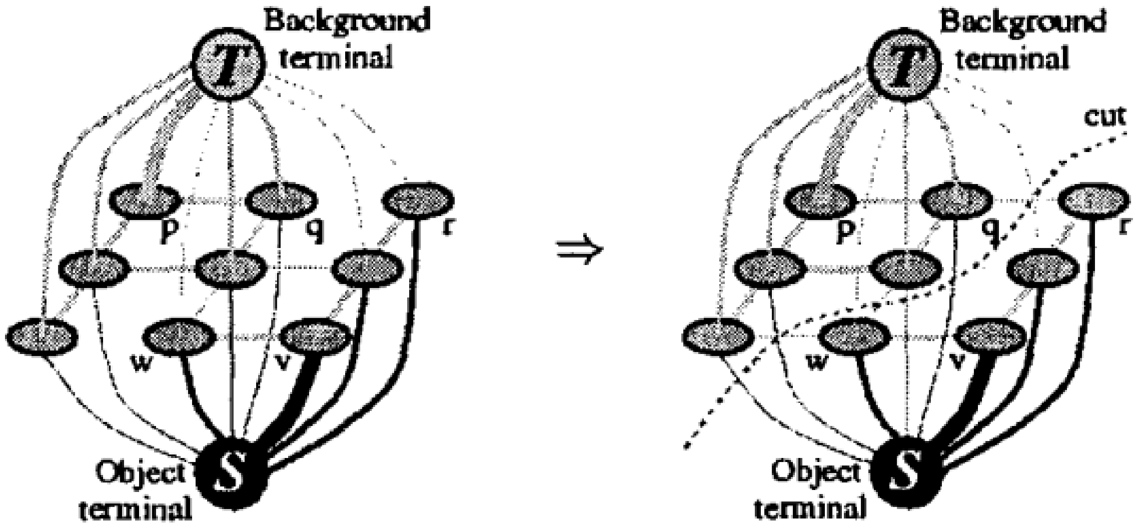
“GrabCut”— Interactive Foreground Extraction using Iterated Graph Cuts

GrabCut

- Grabcut是一种基于图割的交互式分割算法，用户用矩形圈出自己想要分割的前景，grabcut会迭代地将其分割出来，并且迭代过程中可以通过用户交互，对不属于前景的分割结果进行修正。

Graph Cut

- 将整张图片上的每个像素都看作一个节点，相邻像素有边连接。
- 定义源点Source（S）和宿点Sink（T），分别代表前景点和背景点。这两个点与每一个像素结点相连。
- 我们只需要在这张图上跑一遍最小割，就可以将图分为两个独立的子集。与S点相连的作为前景，与T点相连的作为背景。
- 此外，定义能量函数E(A)来评判分割结果



$$E(A) = \lambda \cdot R(A) + B(A) \quad (1)$$

where

$$R(A) = \sum_{p \in \mathcal{P}} R_p(A_p) \quad (2)$$

$$B(A) = \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p, A_q) \quad (3)$$

and

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise.} \end{cases}$$

Graph Cut

- 而在进行最小割算法之前，我们需要为每一条边分配权重。

edge	weight (cost)	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\}\in\mathcal{N}$
$\{p,S\}$	$\lambda\cdot R_p(\text{“bkg”})$	$p\in\mathcal{P},\ p\notin\mathcal{O}\cup\mathcal{B}$
	K	$p\in\mathcal{O}$
	0	$p\in\mathcal{B}$
$\{p,T\}$	$\lambda\cdot R_p(\text{“obj”})$	$p\in\mathcal{P},\ p\notin\mathcal{O}\cup\mathcal{B}$
	0	$p\in\mathcal{O}$
	K	$p\in\mathcal{B}$

而当p本来就是前景时，边权分配为一个极大值，永远不会被分割。
当p本来就是背景时，边权就是0，一开始就会被分割。

N-link

对于{p,q}边，即像素之间相连的边，我们定义为以考虑像素强度对分割结果的影响。
两个像素差越大，这条边的权重也就越小，也就更有可能被分割。

$$B_{\{p,q\}}\propto exp\left(-\frac{(I_p-I_q)^2}{2\sigma^2}\right)\cdot\frac{1}{dist(p,q)}$$

T-link

对于{p,S}边，即像素与前景结点相连的边，我们将权重定义为 $\lambda\cdot R_p(\text{“bkg”})$

$$R_p(\text{“bkg”}) = -\ln\Pr(I_p|\mathcal{B})$$

而Rp(bkg)为p属于背景概率的负对数，Λ为一经验值。
这个概率是通过直方图统计算出来的，等于该像素值在背景直方图中出现的频率。
当p属于背景的概率越大，它与前景结点相连的边权重就越小，就越容易被分割，从而代表它越不可能属于前景。

Graph Cut

- 可以证明，按照上文来定义用于分割的边权，与最初定义的能量函数最小化的目标是等价的。

PROOF: Using the table of edge weights, definition of feasible cuts $\mathcal{F}$, and equation (6) one can show that a cost of any $C \in \mathcal{F}$ is

$$\begin{aligned}|C| &= \sum_{p \notin \mathcal{O} \cup \mathcal{B}} \lambda \cdot R_p(A_p(C)) + \\ &\quad \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p(C), A_q(C)) \\ &= E(A(C)) - \sum_{p \in \mathcal{O}} \lambda R_p(\text{“obj”}) - \sum_{p \in \mathcal{B}} \lambda R_p(\text{“bkg”}).\end{aligned}$$

Therefore, $|C| = E(A(C)) - const(C)$. Note that for any $C \in \mathcal{F}$ assignment $A(C)$ satisfies constraints (4,5). In fact, equation (6) gives a one-to-one correspondence between the set of all feasible cuts in $\mathcal{F}$ and the set $\mathcal{H}$ of all assignments A that satisfy hard constraints (4,5). Then,

$$\begin{aligned}E(\hat{A}) &= |\hat{C}| + const = \min_{C \in \mathcal{F}} |C| + const = \\ &= \min_{C \in \mathcal{F}} E(A(C)) = \min_{A \in \mathcal{H}} E(A)\end{aligned}$$

and the theorem is proved. ■

GrabCut

GrabCut基于GraphCut做了如下改进：

- 为了处理彩色图像，将直方图预测概率变为了GMM预测概率。
- 采用迭代逐步更新模型。
- 通过简易的用户交互进一步校正分割结果。

GrabCut

- 同样的，GrabCut也定义了类似的能量函数，等价于前文中的区域项权值和局部项权值。

of objects. This is captured by a “Gibbs” energy of the form:

$$\mathbf{E}(\underline{\alpha},\underline{\theta},\mathbf{z}) = U(\underline{\alpha},\underline{\theta},\mathbf{z}) + V(\underline{\alpha},\mathbf{z}) \ . \tag{2}$$

The data term U evaluates the fit of the opacity distribution $\underline{\alpha}$ to the data $\mathbf{z}$, given the histogram model $\underline{\theta}$, and is defined to be:

$$U(\underline{\alpha},\underline{\theta},\mathbf{z}) = \sum_n -\log h(z_n;\alpha_n) \, . \tag{3}$$

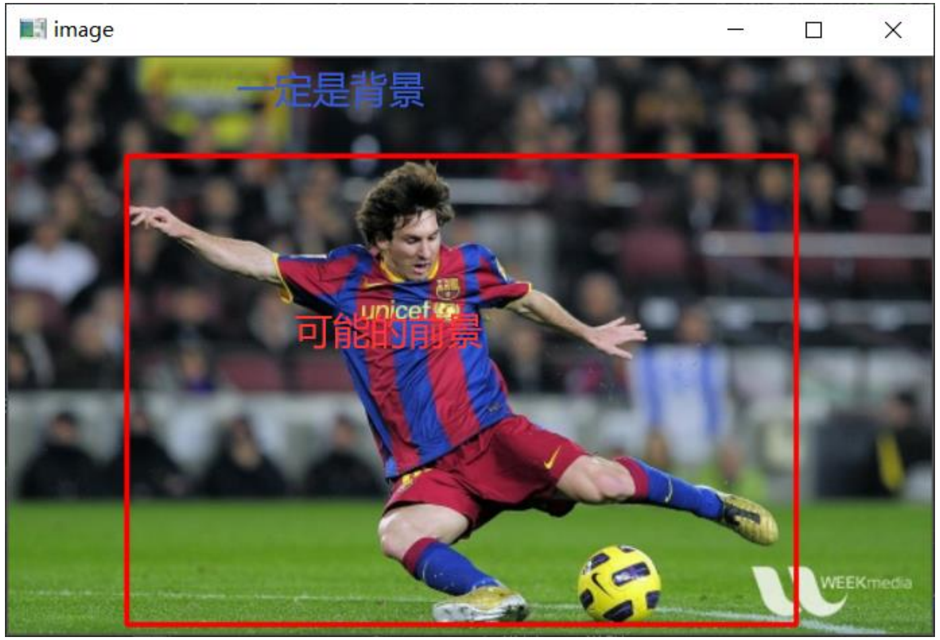
The smoothness term can be written as

$$V(\underline{\alpha},\mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} dis(m,n)^{-1} \, [\alpha_n \neq \alpha_m] \exp -\beta(z_m - z_n)^2, \tag{4}$$

GrabCut算法中认为每个像素可以归为四个类别:

- （确定背景：0， 确定前景：1，可能背景：2，可能前景：3），

所以，用户初始化时，将前景全部纳入一个矩形框内，矩形框以外设置为确定背景，矩形框以内设置为可能前景。



T-link的权重

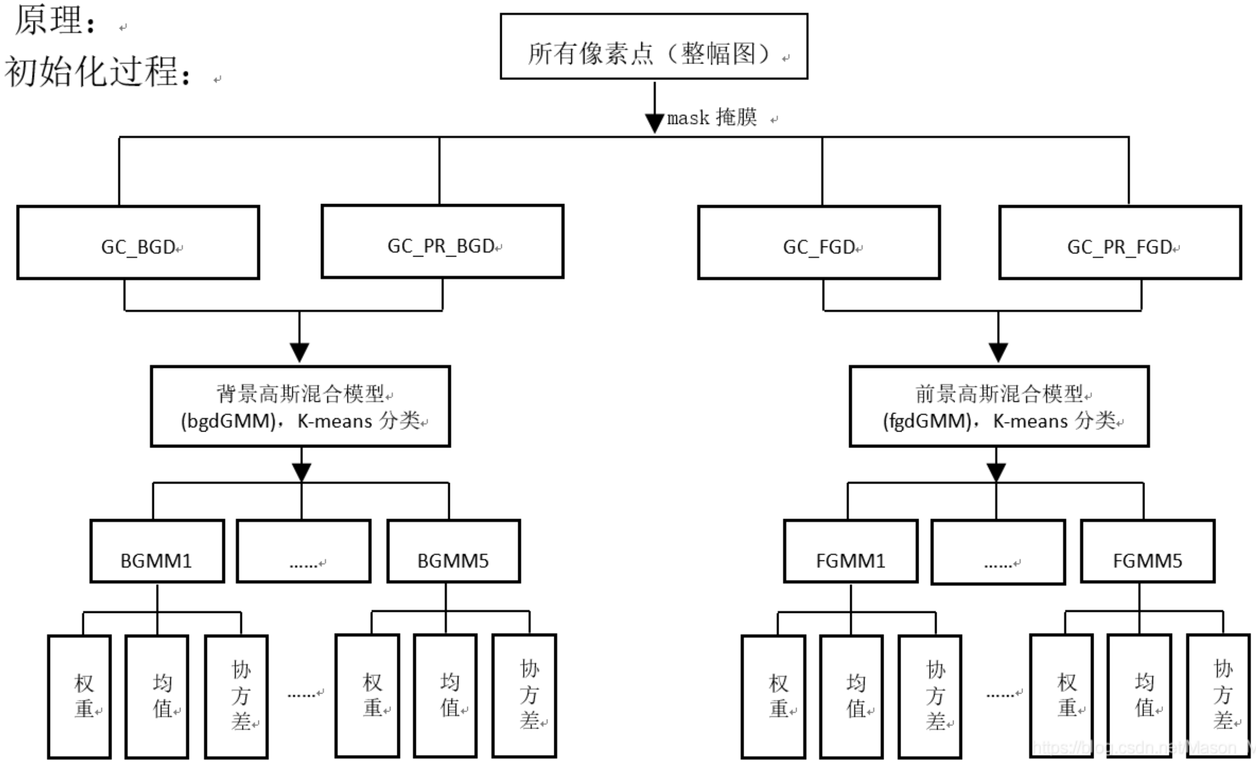
$$U(\underline{\alpha},\mathbf{k},\underline{\theta},\mathbf{z})=\sum_n D(\alpha_n,k_n,\underline{\theta},z_n),$$

$$D(\alpha_n,k_n,\underline{\theta},z_n)=-\log\pi(\alpha_n,k_n)+\frac{1}{2}\log\det\Sigma(\alpha_n,k_n)$$
$$+\frac{1}{2}[z_n-\mu(\alpha_n,k_n)]^\top\Sigma(\alpha_n,k_n)^{-1}[z_n-\mu(\alpha_n,k_n)].$$

- 有了用户提供的先验知识后，就分别为前景和背景建立两套GMM。其中GMM内部的类别为5类。
- 通过EM算法/Kmeans估计两套GMM的参数（权重，均值，协方差）

三、原理：

1. 初始化过程：



N-link的权重

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n)\in \mathbf{C}} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2 .$$

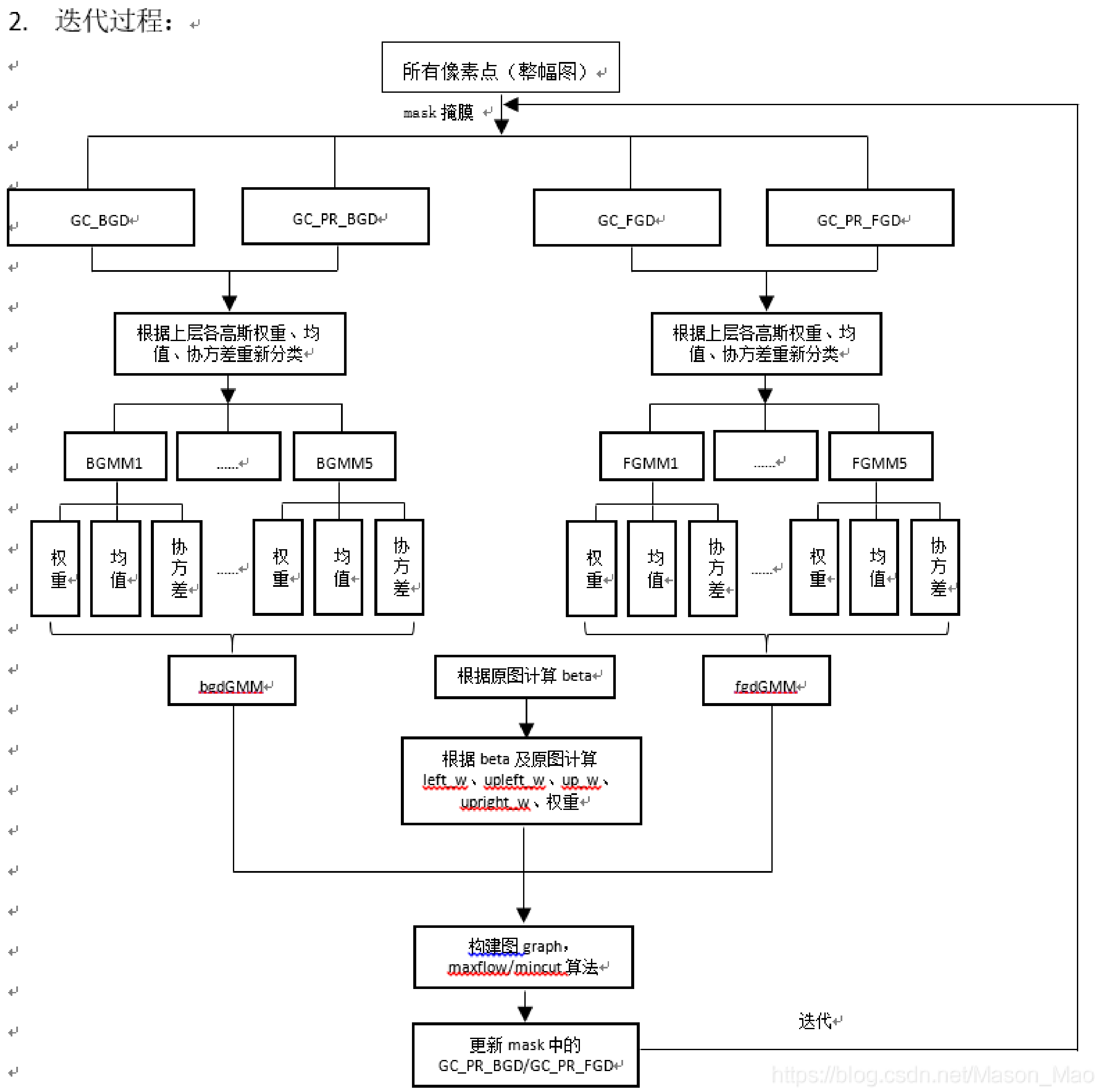
- GrabCut做了一些小改动。因为GrabCut处理的是彩色图像，所以衡量像素差是通过L2范数实现的。
- 此外，为了适应不同对比度的图像，需要乘以一个β来对像素之间的差进行调整。这个β值是在图像全局上计算出来的。

nectivity). When the constant $\beta = 0$, the smoothness term is simply the well-known Ising prior, encouraging smoothness everywhere, to a degree determined by the constant γ. It has been shown however [Boykov and Jolly 2001] that it is far more effective to set $\beta > 0$ as this relaxes the tendency to smoothness in regions of high contrast. The constant β is chosen [Boykov and Jolly 2001] to be:

$$\beta = \left(2\left\langle (z_m - z_n)^2 \right\rangle\right)^{-1}, \tag{5}$$

where $\langle\cdot\rangle$ denotes expectation over an image sample.

- 每次迭代，像素的归属（前景/背景）会发生变动，GMM需要重新拟合数据。



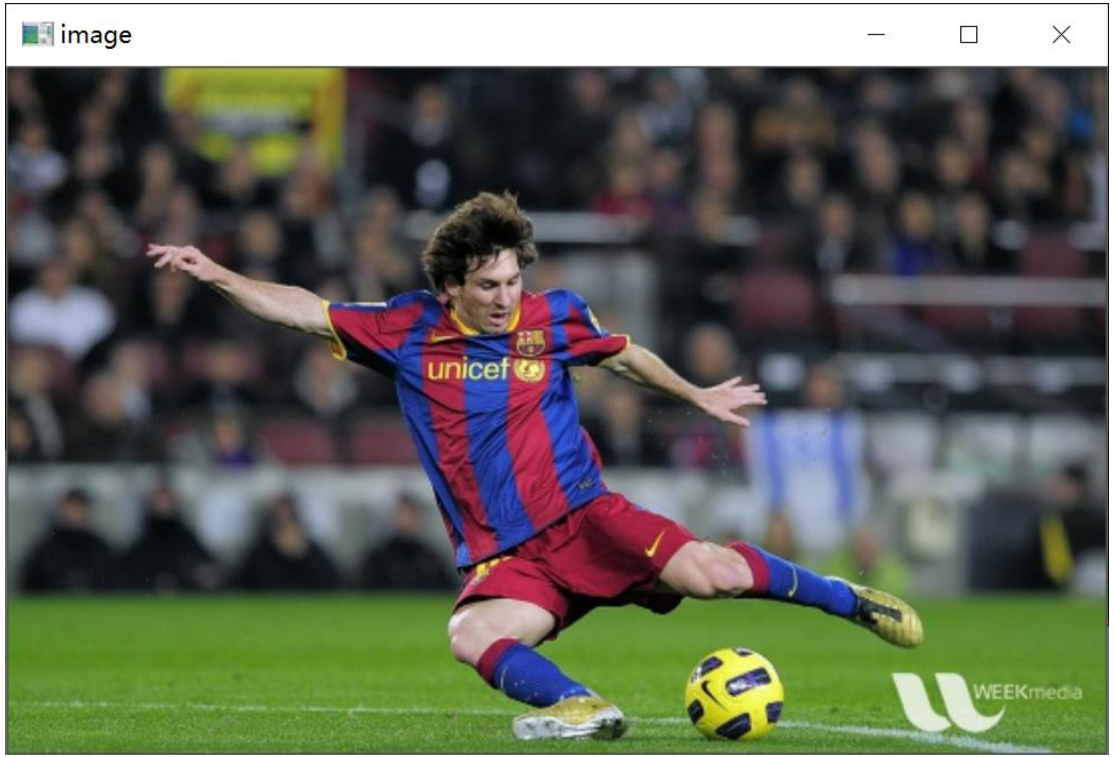


image
unicef
WEEKmedia

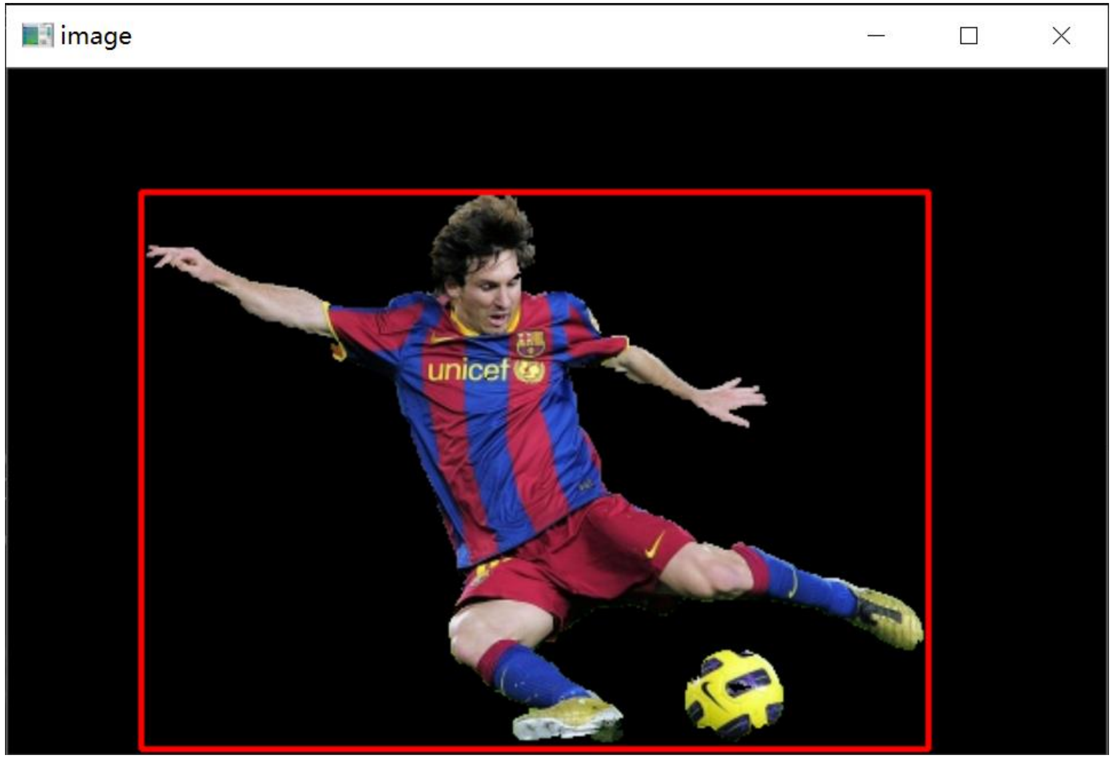


image
unicef