Lecture 13: Applications of structured output learning

Boris Flach and Vojtěch Franc

May 21, 2015

11.A: Number Plate Recognition

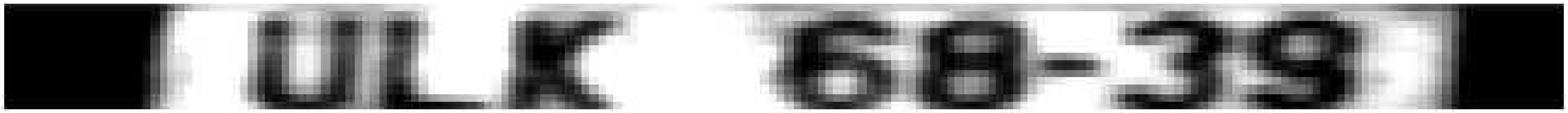
11.B: Facial Landmark Detector

11.C: Models & generative learning tasks for segmentation

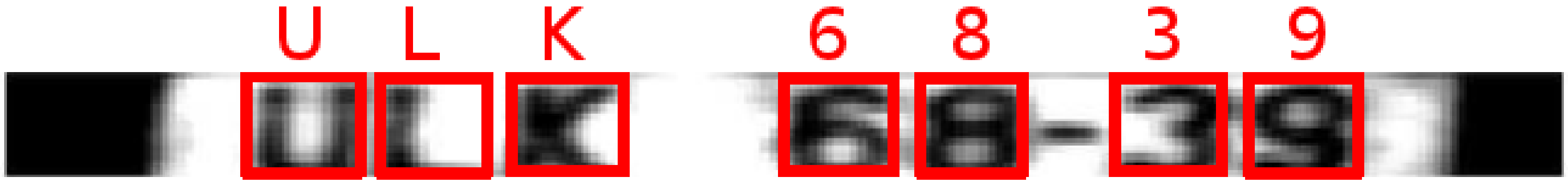
XEP33SML – Structured Model Learning, Summer 2015

11.A: Number Plate Recognition

- Task: given an image containing a line of text



  we want to recognize the characters:



- The classical approach is to split the problem into two steps:

  1. Segmentation: find the positions of the characters in the image
  2. Recognition: run OCR on sub-windows found by the segmentation step

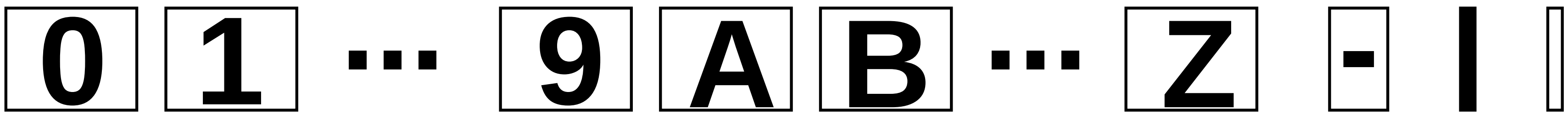
  A problem of the two step approach is that for a good segmentation you should know a model of the sought object.

- Can we solve the segmentation and the recognition problem simultaneously?

11.A: Number Plate Recognition

- The structured classifier input is an image $\boldsymbol{x} \in \mathcal{X}$ of size $[H \times W]$.

- The input image $\boldsymbol{x}$ can be modeled as a sequence of templates selected from a finite set of templates $\boldsymbol{w} = \{\boldsymbol{w}_a \in \mathbb{R}^{H \times \omega(a)} | a \in \mathcal{A}\}$



- The classifier output is the image segmentation $\boldsymbol{y} = (s_1, \ldots, s_L)$ where each segment $s = (a, k)$ describes a character name $a \in \mathcal{A}$ and its position $k \in \{1, \ldots, W\}$.

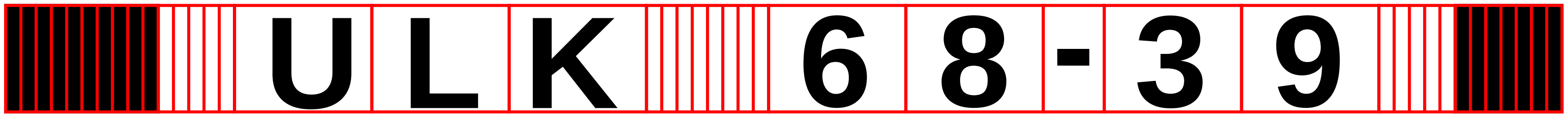
- An admissible segmentation $\boldsymbol{y} = (s_1, \ldots, s_L) \in \mathcal{Y}$ defines a sequence of non-overlapping templates covering the whole image $\boldsymbol{x} \in \mathbb{R}^{H \times W}$, i.e.

$$
\begin{aligned}
k(s_1) &= 1 \\
W &= k(s_L) + \omega(s_L) - 1 \\
k(s_i) &= k(s_{i-1}) + \omega(s_{i-1}) \,, \; i = 2, \ldots, L
\end{aligned}
$$

where $k(s_i)$ is the position and $\omega(s_i)$ is the width of the i-th segment.

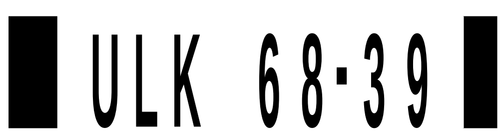
11.A: Number Plate Recognition

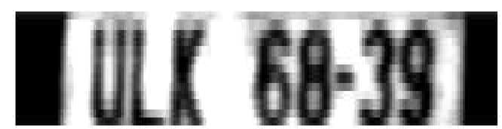
- Given an admissible segmentation $\boldsymbol{y} \in \mathcal{Y}$ and the templates $\boldsymbol{w} = \{\boldsymbol{w}_a \in \mathbb{R}^{H \times \omega(a)} | a \in \mathcal{A}\}$ we can generate synthetic image:



- The similarity between the input image $\boldsymbol{x} \in \mathbb{R}^{H \times W}$ and a synthetic image generated from $\boldsymbol{w}$ and $\boldsymbol{y} = (s_1, \ldots, s_L) \in \mathcal{Y}$ can be measured by their correlation:

$$f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w}) = \underbrace{\sum_{i=1}^{L(y)} \sum_{j=1}^{\omega(a(s_i))} \langle \mathrm{col}(\boldsymbol{x}, j + k(s_i) - 1), \mathrm{col}(\boldsymbol{w}_{a(s_i)}, j) \rangle}$$

$\langle$ 

, 

$\rangle$

- The best segmentation can be found by finding among all admissible synthetic images the one having the highest correlation with the input

$$\hat{y} \in \arg\max_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle$$

which is a linear classifier and $\boldsymbol{\Psi} \colon \mathbb{R}^{H \times W} \times \mathcal{Y} \to \mathbb{R}^n$ is a properly designed feature map.

11.A: Number Plate Recognition

- The maximization task needed to evaluate the linear classifier

$$\max_{\boldsymbol{y}\in\mathcal{Y}}\left[\sum_{i=1}^{L(y)}\sum_{j=1}^{\omega(a(s_i))}\langle \mathrm{col}(\boldsymbol{x}, j+k(s_i)-1), \mathrm{col}(\boldsymbol{w}_{a(s_i)}, j)\rangle\right]$$

  can be solved by the dynamic programming.

- For $i=1$ to W compute

$$f_i = \max_{a\in\mathcal{A},\omega(a)\leq i}\left[\sum_{j=1}^{\omega(a)}\langle \mathrm{col}(\boldsymbol{x}, i-\omega(a)+j), \mathrm{col}(\boldsymbol{w}_a, j)\rangle + f_{i-\omega(a)}\right]$$

  and then read the maximizing segmentation in the backward direction.

- The classifier can be equivalently formulated as an instance of the max-sum classifier with a chain neighborhood structure.

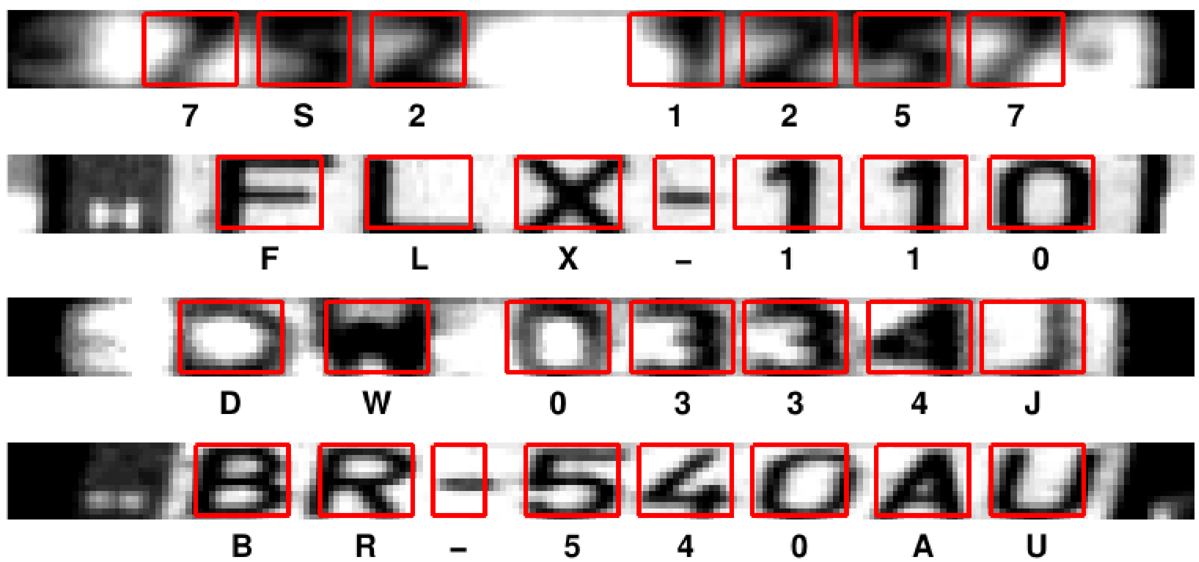
11.A: Number Plate Recognition

- **Learning task:** find the templates $\boldsymbol{w} = \{\boldsymbol{w}_a \in \mathbb{R}^{H\times\omega(a)} | a \in \mathcal{A}\}$ to minimize the expected risk with the loss function $\ell\colon \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$ defined as follows:

$$\ell(\boldsymbol{y},\boldsymbol{y}') = \frac{1}{W}\sum_{i=1}^{W} [\![a(\boldsymbol{y},i) \neq a(\boldsymbol{y}',i)]\!]$$

where $a\colon \mathcal{Y}\times\{1,\ldots,W\}\to\mathcal{A}$ returns a name of character covering the i-th column

- Learning of the templates $\boldsymbol{w}$ from manually annotated examples

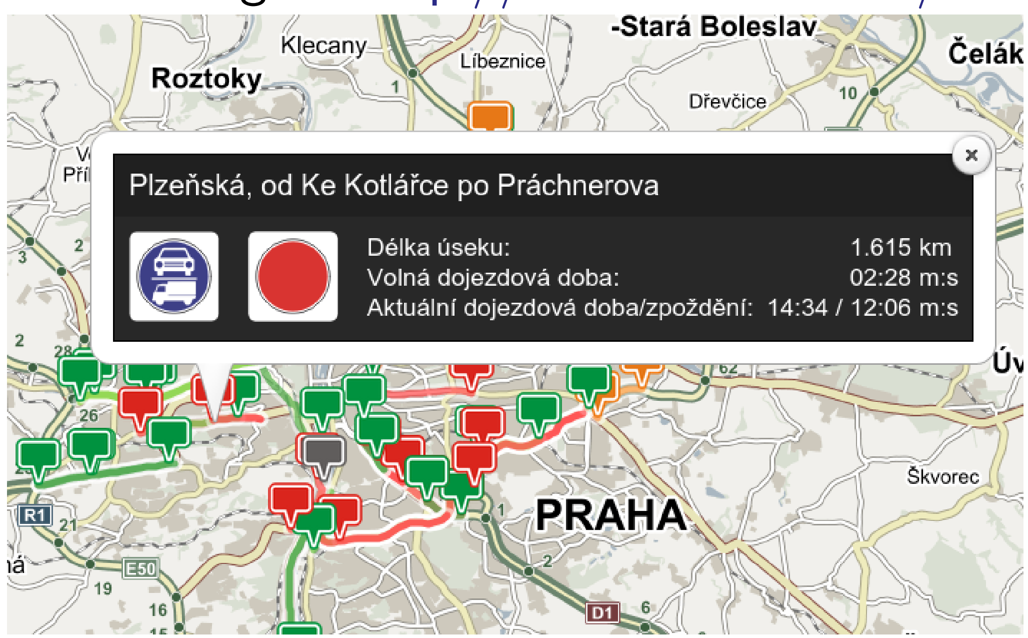


formulated as the regularized risk minimization with the margin-rescaling proxy:

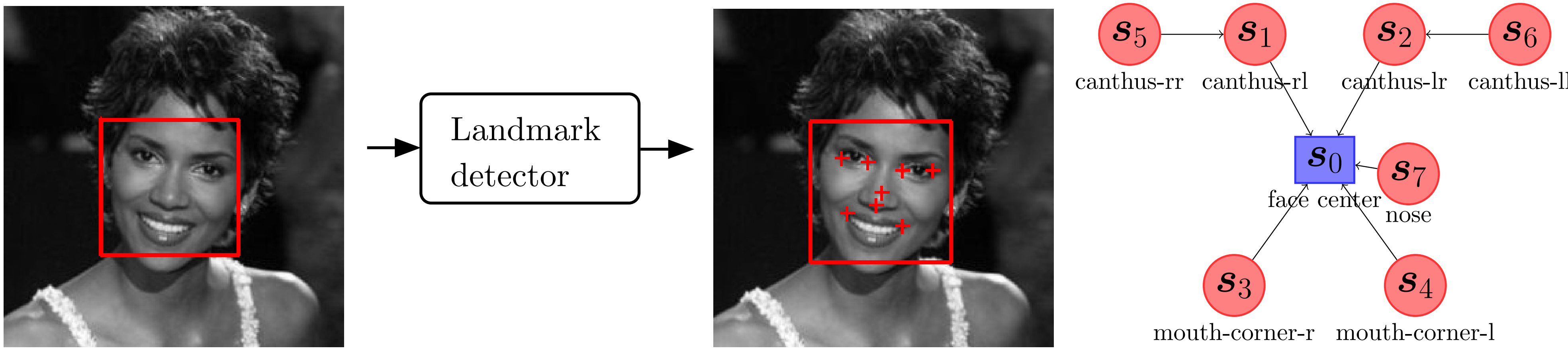
$$\boldsymbol{w}^* = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^n} \left[\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \max_{\boldsymbol{y}\in\mathcal{Y}} \left[\ell(\boldsymbol{y}^i,\boldsymbol{y}) - \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y}^i)\rangle + \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y})\rangle \right] \right]$$

11.A: Number Plate Recognition

Travel time in Prague: http://unicam.camea.cz/Discoverer/



11.B: Facial Landmark Detection



A structured output classifier (landmark detector) simultaneously estimates the viewing angle and the positions of facial landmarks:

$$(\hat{\phi}, \hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_{L^{\hat{\phi}}}) = \mathop{\arg\max}_{\substack{\phi \in \Phi \\ (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{L^\phi}) \in \mathbb{N}^{2 \times L}}} \Big[f_\phi(\boldsymbol{x}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{L^\phi}; \boldsymbol{w}) + b_\phi(\boldsymbol{w}) \Big]$$

where the score of each view $\phi \in \Phi$ is a deformable part model (DPM):

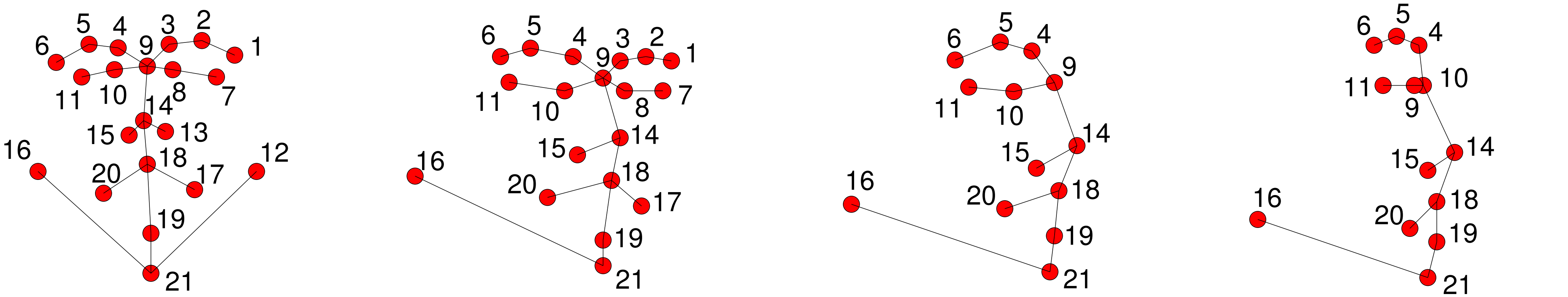
$$f_\phi(\boldsymbol{x}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{L^\phi}; \boldsymbol{w}) = \underbrace{\sum_{v \in \mathcal{V}^\phi} f_v^\phi(\boldsymbol{x}, \boldsymbol{s}_v; \boldsymbol{w})}_{\text{match with the image}} + \underbrace{\sum_{vv' \in \mathcal{E}^\phi} f_{vv'}^\phi(\boldsymbol{s}_v, \boldsymbol{s}_{v'}; \boldsymbol{w})}_{\text{shape prior}}$$

11.B: Facial Landmark Detection

- Each view $\phi \in \Phi$ has its own DPM model with different neighboring structure:

$$f_\phi(\boldsymbol{x}, \phi, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{L^\phi}; \boldsymbol{w}) = \sum_{v \in \mathcal{V}^\phi} f_v^\phi(\boldsymbol{x}, \boldsymbol{s}_v; \boldsymbol{w}) + \sum_{vv' \in \mathcal{E}^\phi} f_{vv'}^\phi(\boldsymbol{s}_v, \boldsymbol{s}_{v'}; \boldsymbol{w})$$

frontal $(-15^\circ, 15^\circ)$ semi-profile $(15^\circ, 45^\circ)$ semi-profile $(45^\circ, 75^\circ)$ profile $(75^\circ, 90^\circ)$



- The unary potential $f_v^\phi(\boldsymbol{x}, \boldsymbol{s}_v; \boldsymbol{w}) = \langle \boldsymbol{w}_v^\phi, \boldsymbol{\Psi}_v^\phi(\boldsymbol{x}, \boldsymbol{s}_v) \rangle$ is a dot product between $\boldsymbol{w}_v^\phi$ and features $\boldsymbol{\Psi}_v^\phi(\boldsymbol{x}, \boldsymbol{s}_v)$ computed on a sub-image cropped from $\boldsymbol{x}$ around the position $\boldsymbol{s}_v$.
- The pair-wise potential $f_{vv'}^\phi(\boldsymbol{s}_v, \boldsymbol{s}_{v'}; \boldsymbol{w}) = \langle \boldsymbol{w}_{vv'}^\phi, \boldsymbol{\Psi}_{vv'}^\phi(\boldsymbol{s}_v, \boldsymbol{s}_{v'}) \rangle$ is a score assigned to a displacement vector $\boldsymbol{s}_v - \boldsymbol{s}_{v'}$ of the neighboring landmarks $\{v, v'\} \in \mathcal{E}$.

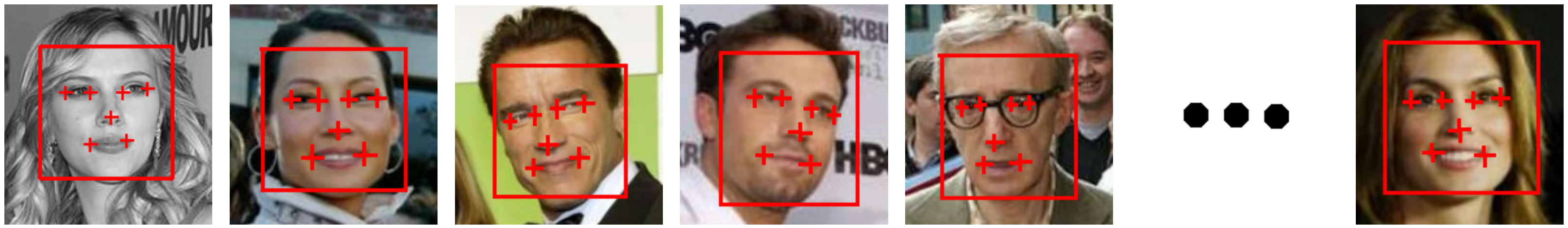
11.B: Facial Landmark Detection

- The loss function penalizes deviations in the landmark positions as well as the estimated viewing angle:

$$\ell\big((\phi, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_L), (\phi', \boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_L)\big) = \begin{cases} \frac{1}{\kappa(\boldsymbol{s}_1,\ldots,\boldsymbol{s}_L)L} \sum_{v\in\mathcal{V}} \|\boldsymbol{s}_v - \boldsymbol{s}'_v\| & \text{if } \phi = \phi' \\ 1 & \text{otherwise} \end{cases}$$

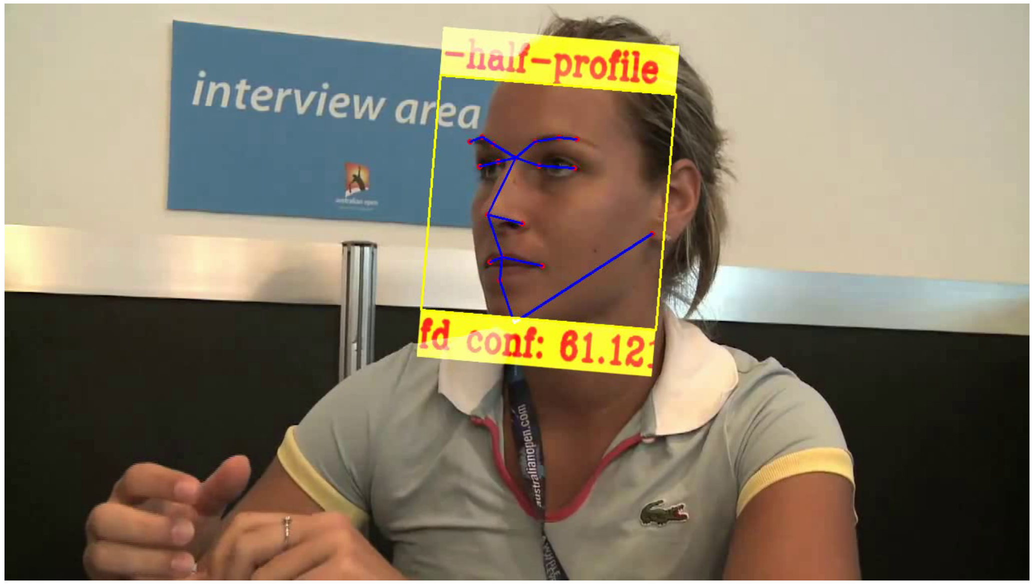
where $\kappa(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L)$ is the face size defined by the ground truth positions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L$.

- **Learning task:** find the parameters $\boldsymbol{w}$ from the training set composed of manually annotated faces $\{(\boldsymbol{x}^1, \phi^1, \boldsymbol{s}^1_1, \ldots, \boldsymbol{s}^1_{L_{\phi^1}}), \ldots, (\boldsymbol{x}^m, \phi^m, \boldsymbol{s}^m_1, \ldots, \boldsymbol{s}^m_{L_{\phi^m}})\}$



by the regularized risk minimization with the margin-rescaling proxy.

11.A: Facial Landmark Detection



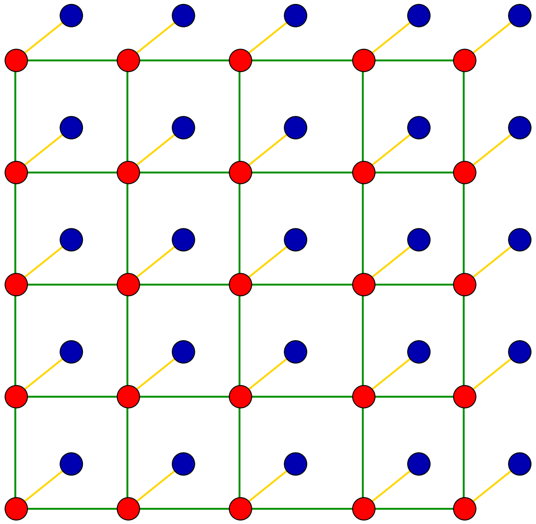
- An open-source library implementing the landmark detector and its learning developed by Michal Uricar is downloadable here.

11.C: Models & generative learning tasks for segmentation

Simple homogeneous Gibbs random field

$$p(\mathbf{x}, \mathbf{s}) = \frac{1}{Z(u, v)} \exp\Big[\sum_{ij \in E_1} u(s_i, s_j) + \sum_{ij \in E_2} v(s_i, x_j)\Big]$$

where $\mathbf{x}$ - image, s - segmentation and $s_i \in K$, $x_i \in F$.
Gibbs potentials for edges in E_1 fixed, e.g. Potts model

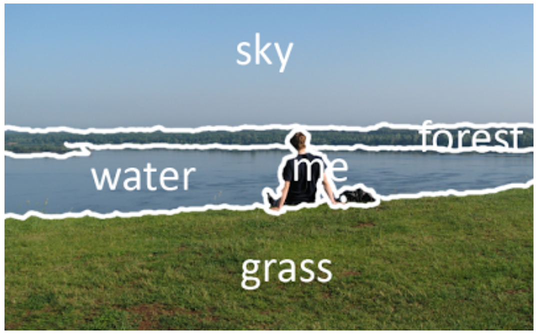


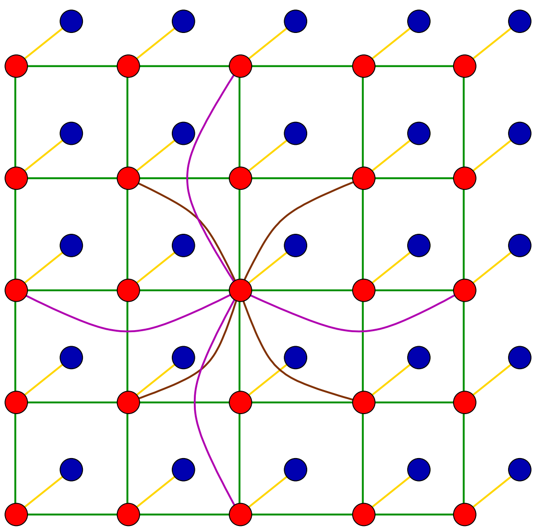
$$u(s_i, s_j) = \begin{cases} 0 & \text{if } s_i = s_j, \\ -\alpha & \text{otherwise.} \end{cases}$$

Gibbs potentials for edges in E_2 (appearance model) should be learned.

- supervised case $\Rightarrow$ “trivial”
- unsupervised case $\Rightarrow$ EM algorithm, requires computation of posterior marginal probabilities $p(s_i \mid \mathbf{x})$, hard.

Homogeneous Gibbs random field





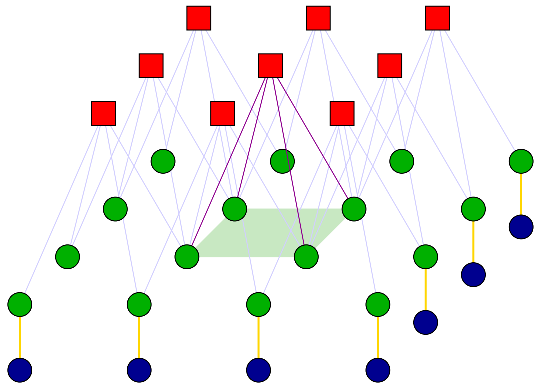
$$p(\mathbf{x}, \mathbf{s}) = \frac{1}{Z(u, v)} \exp\Big[\sum_{ij \in E_1} u_1(s_i, s_j) + \sum_{ij \in E_2} u_2(s_i, s_j) + \ldots + \sum_{ij \in E_m} v(s_i, x_j) \Big]$$

where $\mathbf{x}$ - image, $\mathbf{s}$ - segmentation and $s_i \in K$, $x_i \in F$.
Learn all Gibbs potentials $u_1, u_2 \ldots$ and appearance model v.

- supervised case $\Rightarrow$ use pseudo-likelihood maximisation,
- unsupervised case $\Rightarrow$ EM algorithm, requires computation of posterior/prior unary/pairwise marginal probabilities $p(s_i, s_j \mid \mathbf{x}), \ldots$, hard.

Homogeneous Gibbs random field with latent variables

$$p(\mathbf{x}, \mathbf{s}, \mathbf{y}) = \frac{1}{Z(u, v)} \exp\Big[\sum_{ij \in E_1} u_1(s_i, y_j) + \ldots + \sum_{ij \in E_m} v(s_i, x_j)\Big]$$



where $\mathbf{x}$ - image, $\mathbf{s}$ - segmentation, $\mathbf{y}$ - latent variables and $s_i \in K$, $x_i \in F$, $y_i \in M$.

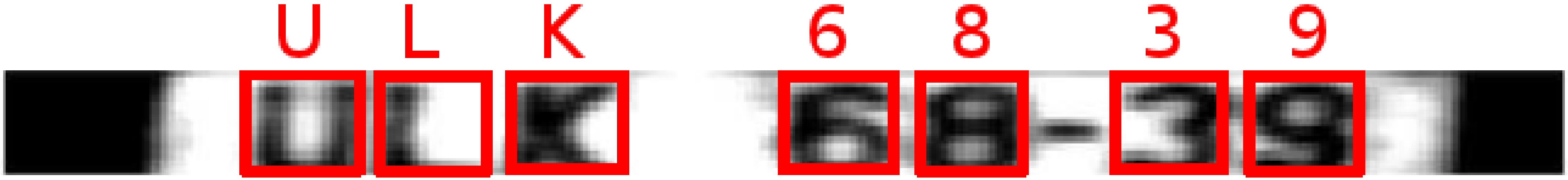
- Potentials $u_1, \ldots$ define a complex shape model

$$p(\mathbf{s}) \sim \sum_{\mathbf{y}} \exp\Big[\sum_{ij \in E_1} u_1(s_i, y_j) + \ldots\Big]$$

i.e. via marginalisation over the latent variables $\mathbf{y}$.

- Requires unsupervised learning even if training data $\mathcal{T} = \{(\mathbf{x}^\ell, \mathbf{s}^\ell) \mid \ell = 1, \ldots, L\}$ are given; is hard.

ULK 68-39

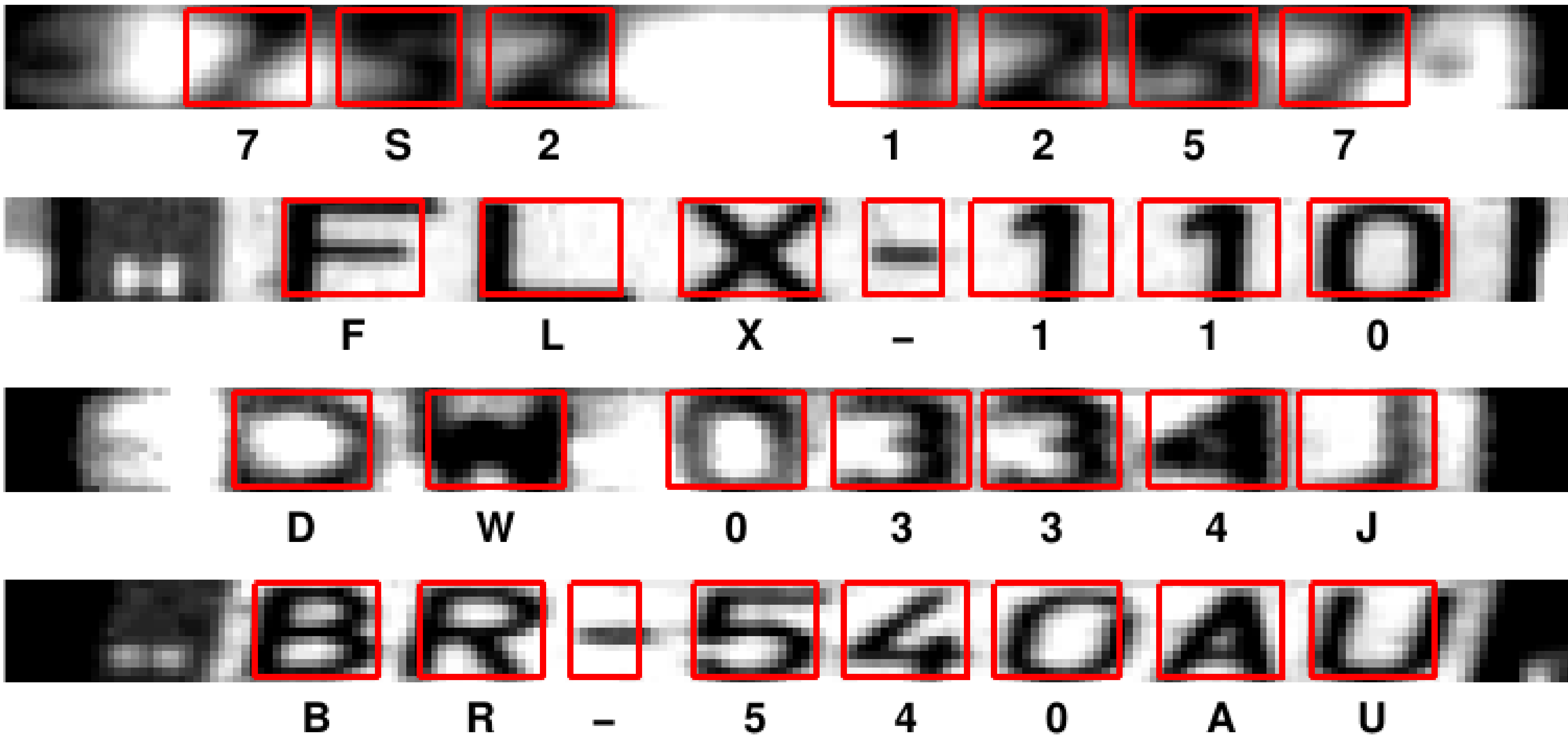

U L K 6 8 3 9

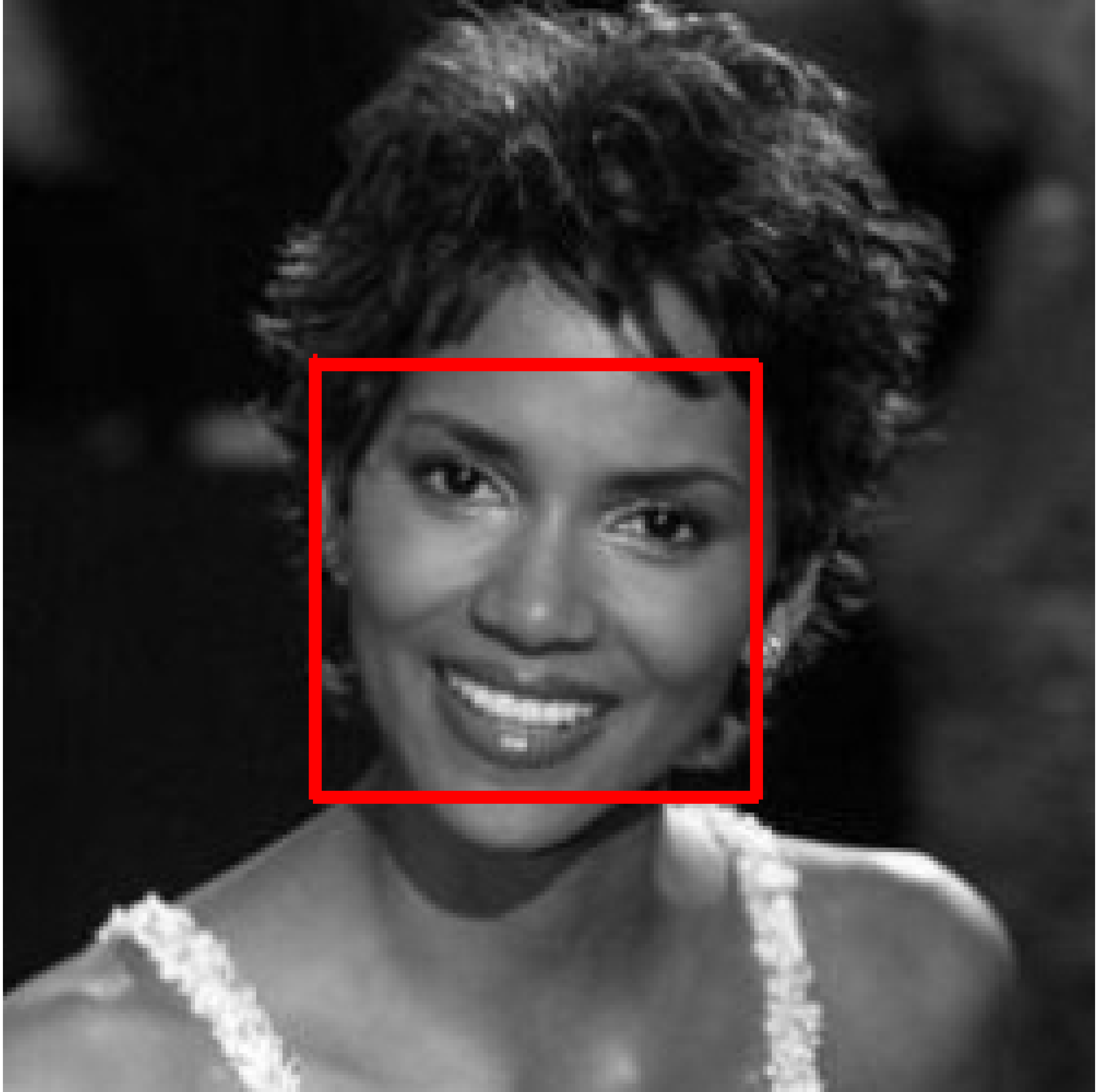
0 1 ··· 9 A B ··· Z - | |

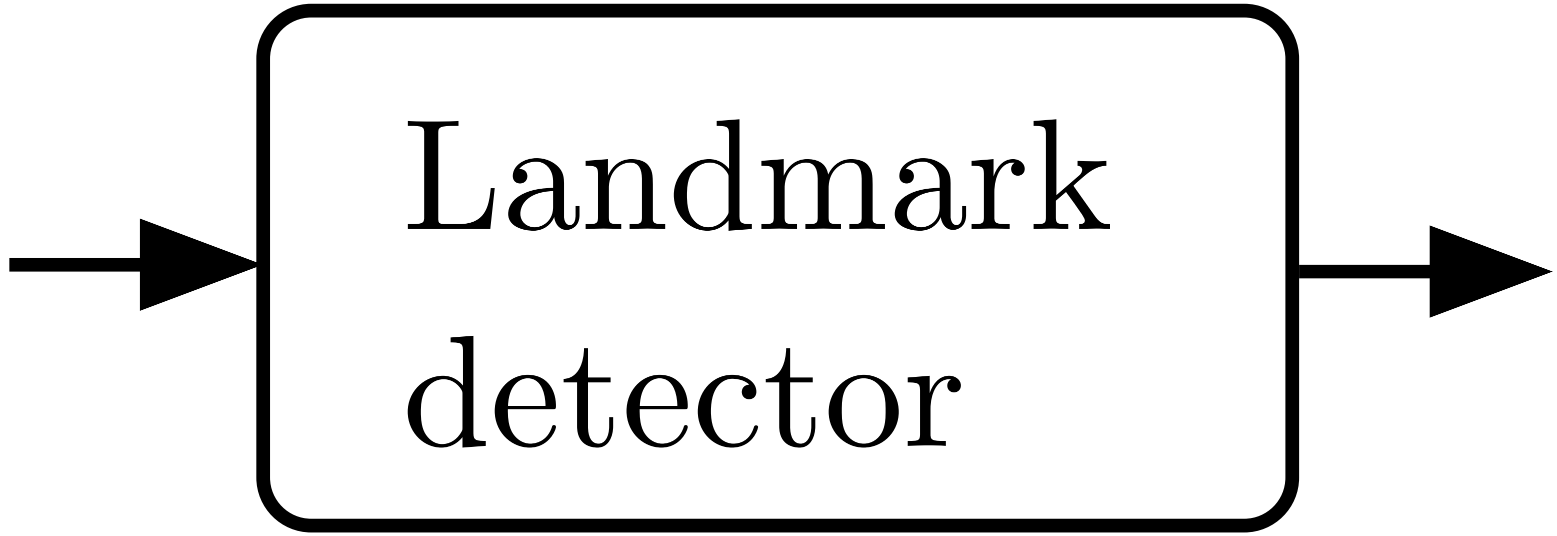
ULK 68-39

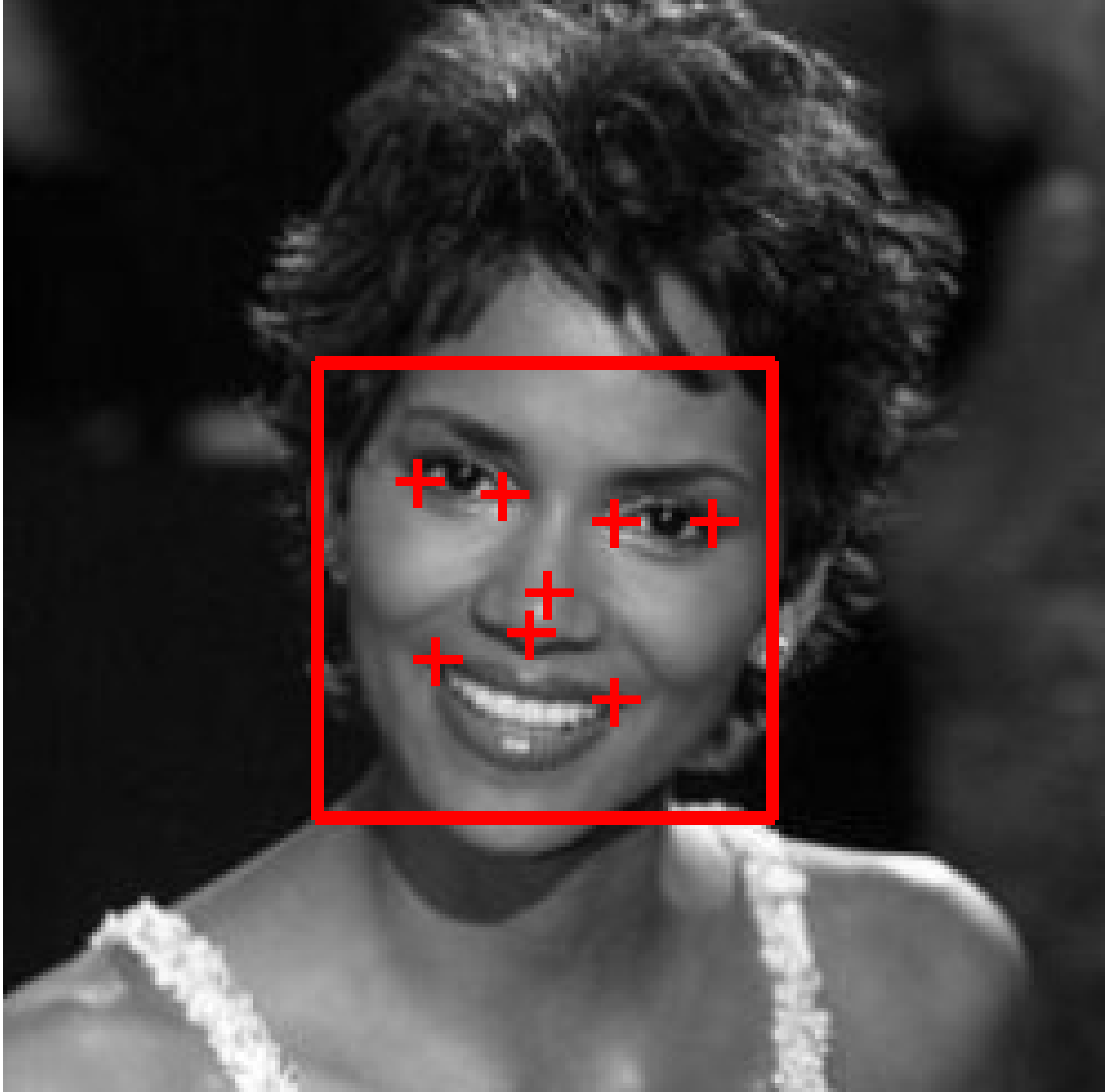
ULK 68-39

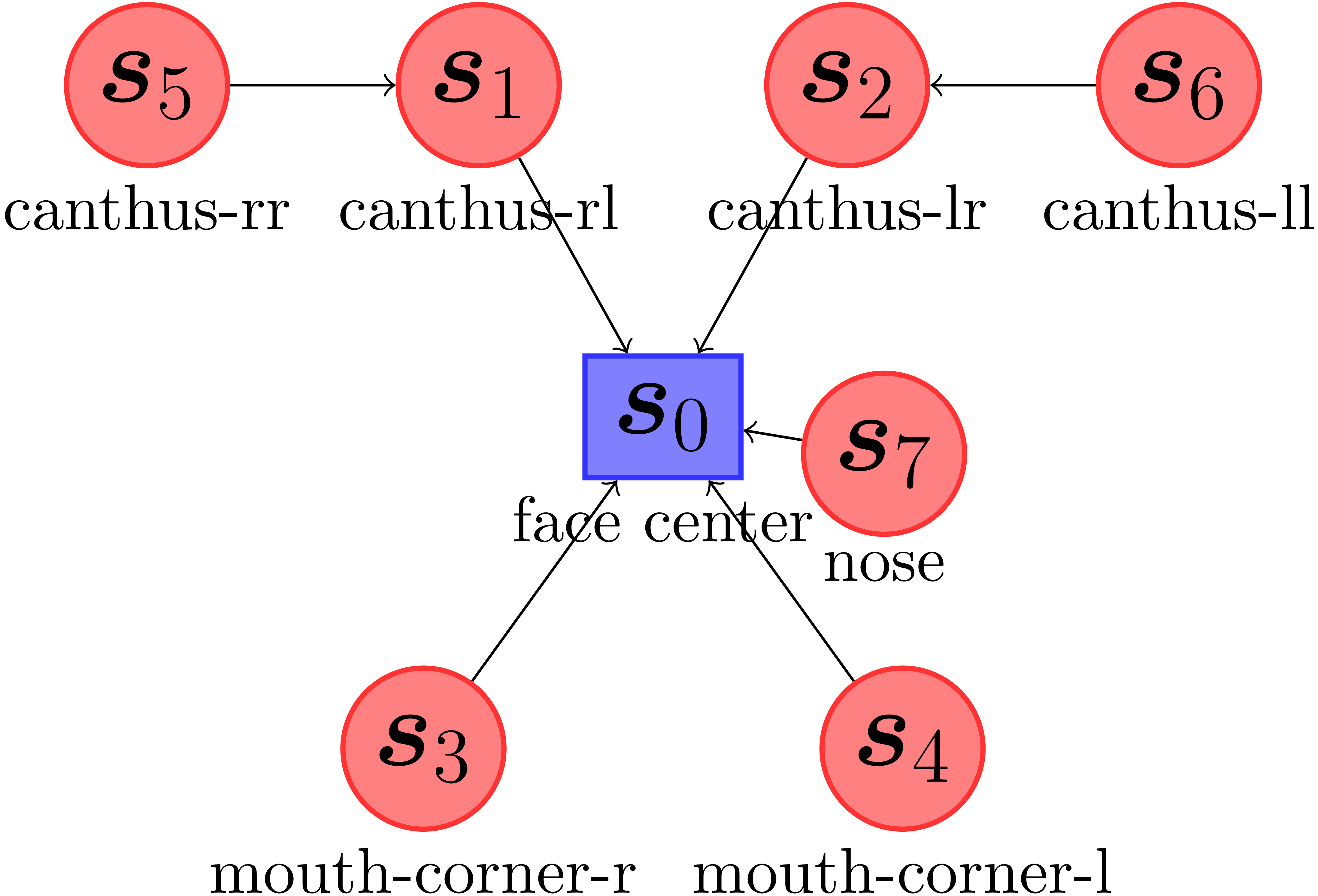
ULK 68-39

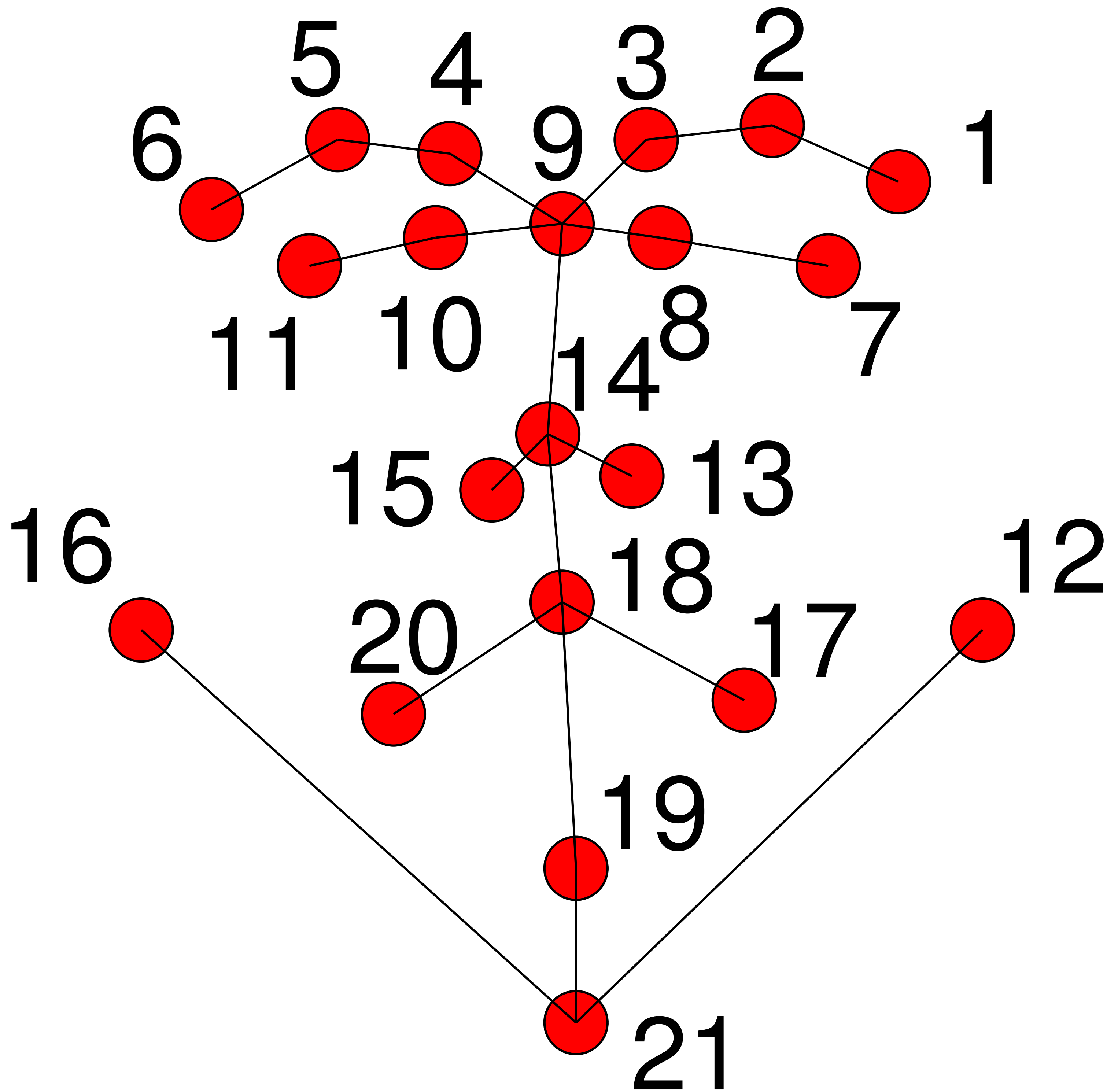

7 S 2 1 2 5 7
F L X – 1 1 0
D W 0 3 3 4 J
B R – 5 4 0 A U



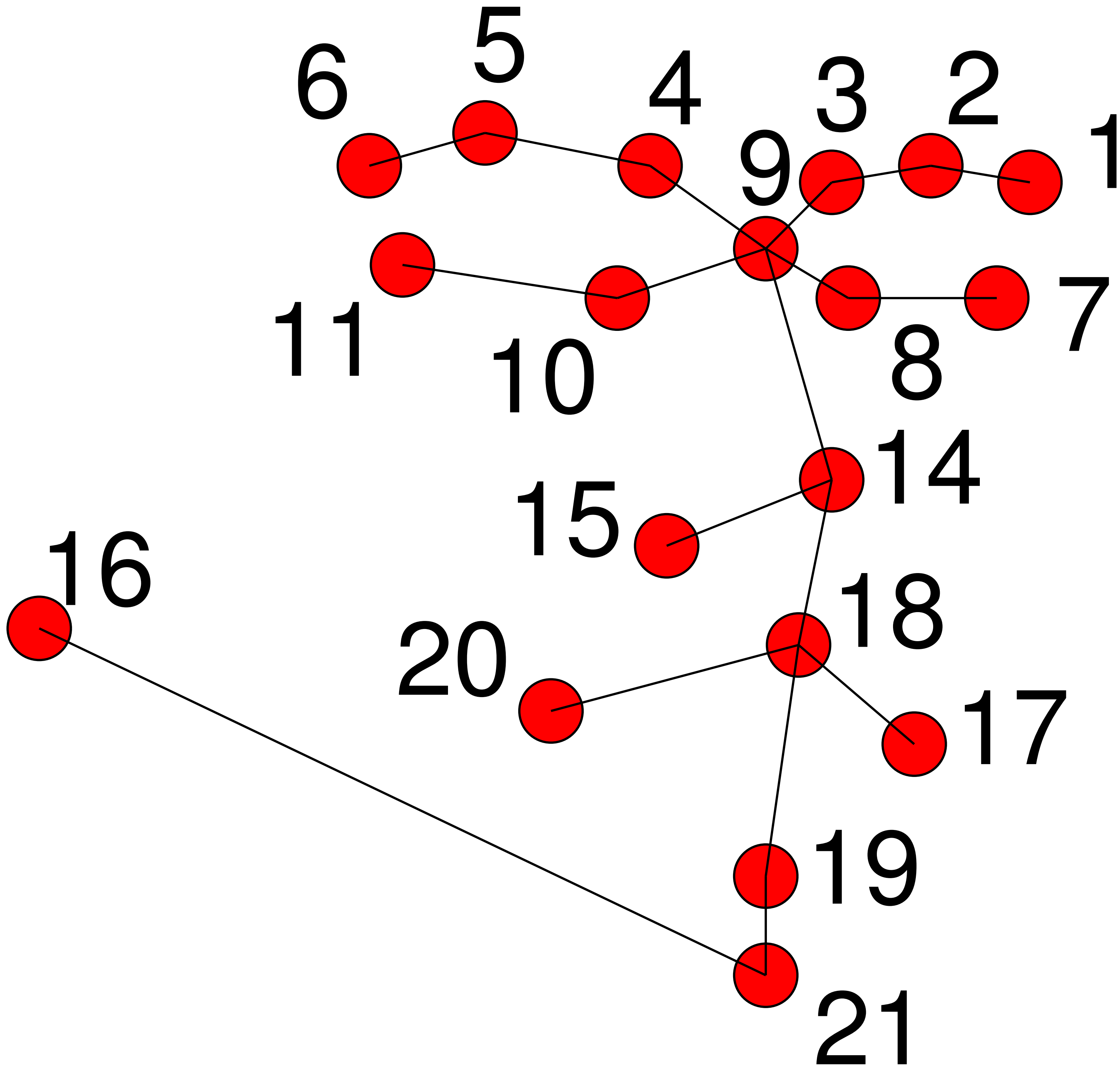

Landmark
detector



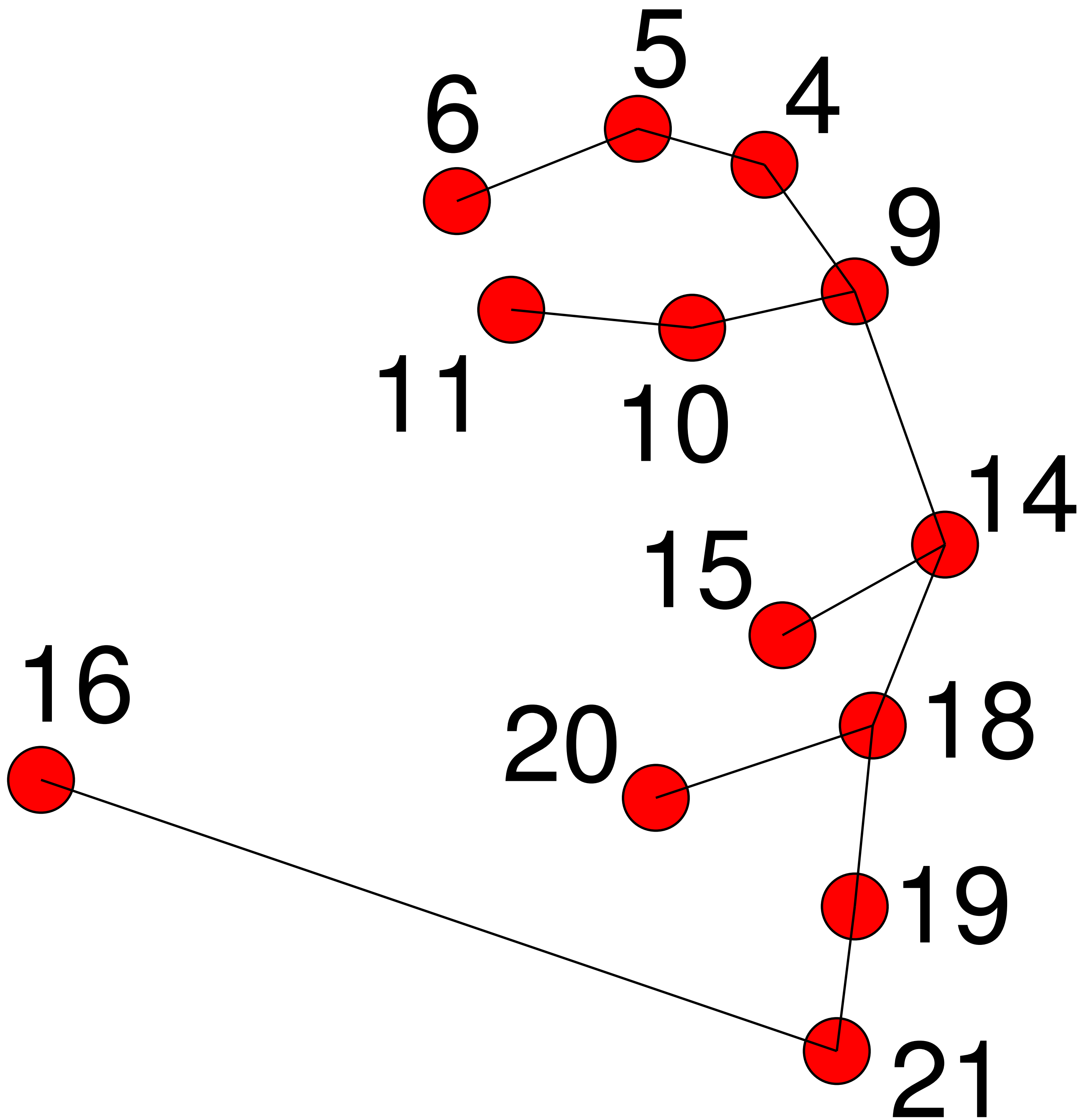

s_5
s_1
s_2
s_6
canthus-rr
canthus-rl
canthus-lr
canthus-ll
s_0
s_7
face center
nose
s_3
s_4
mouth-corner-r
mouth-corner-l



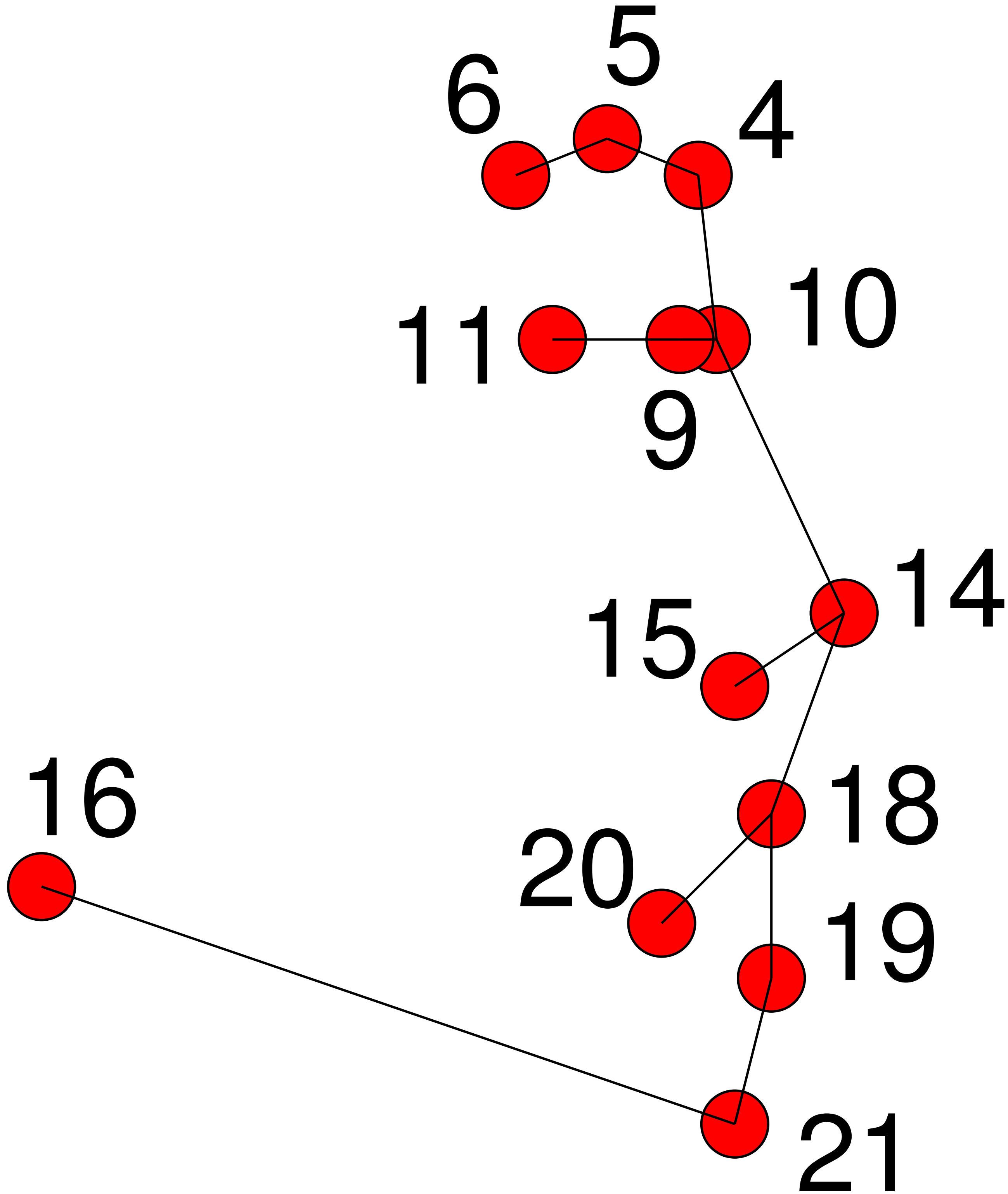
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

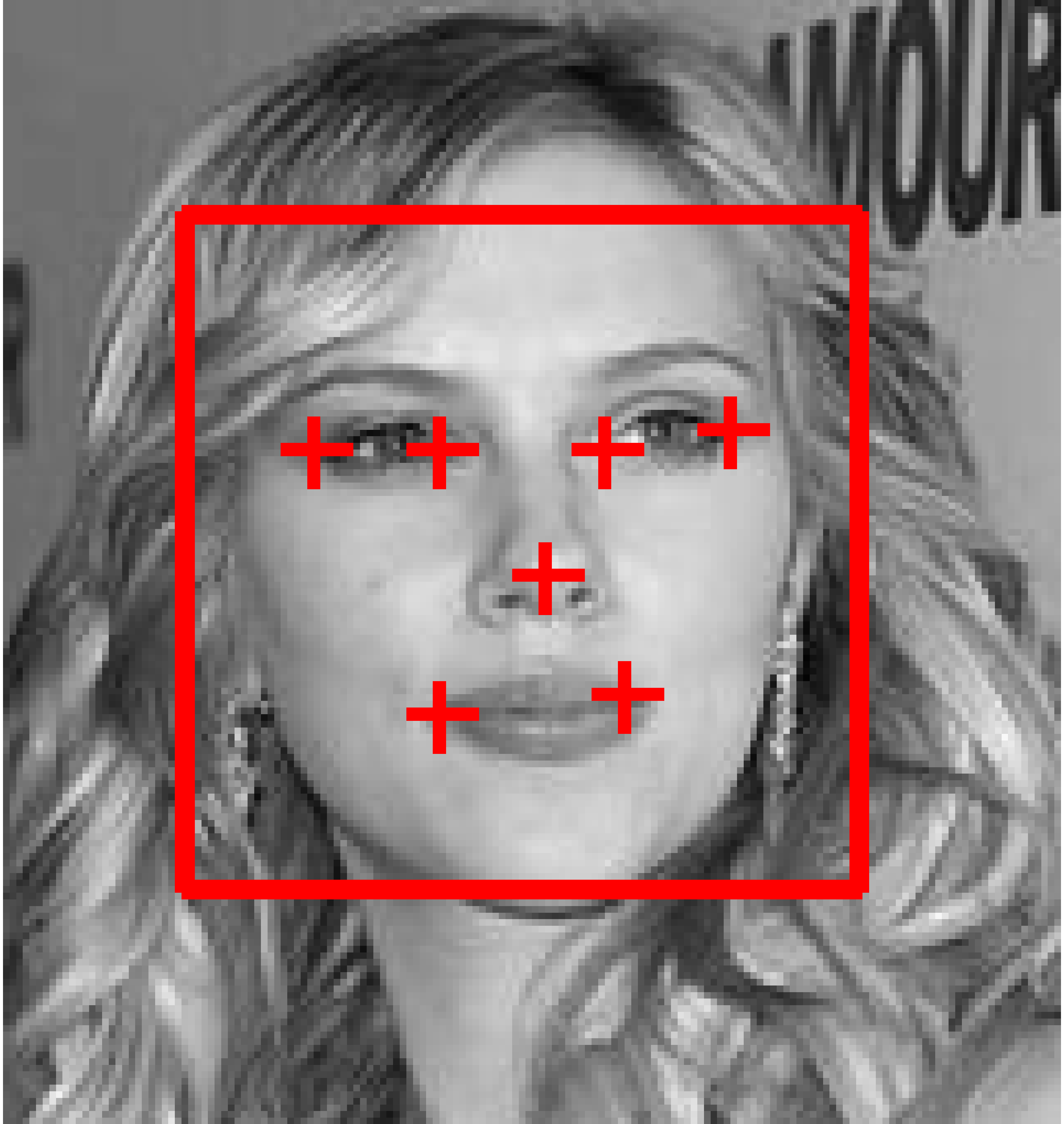


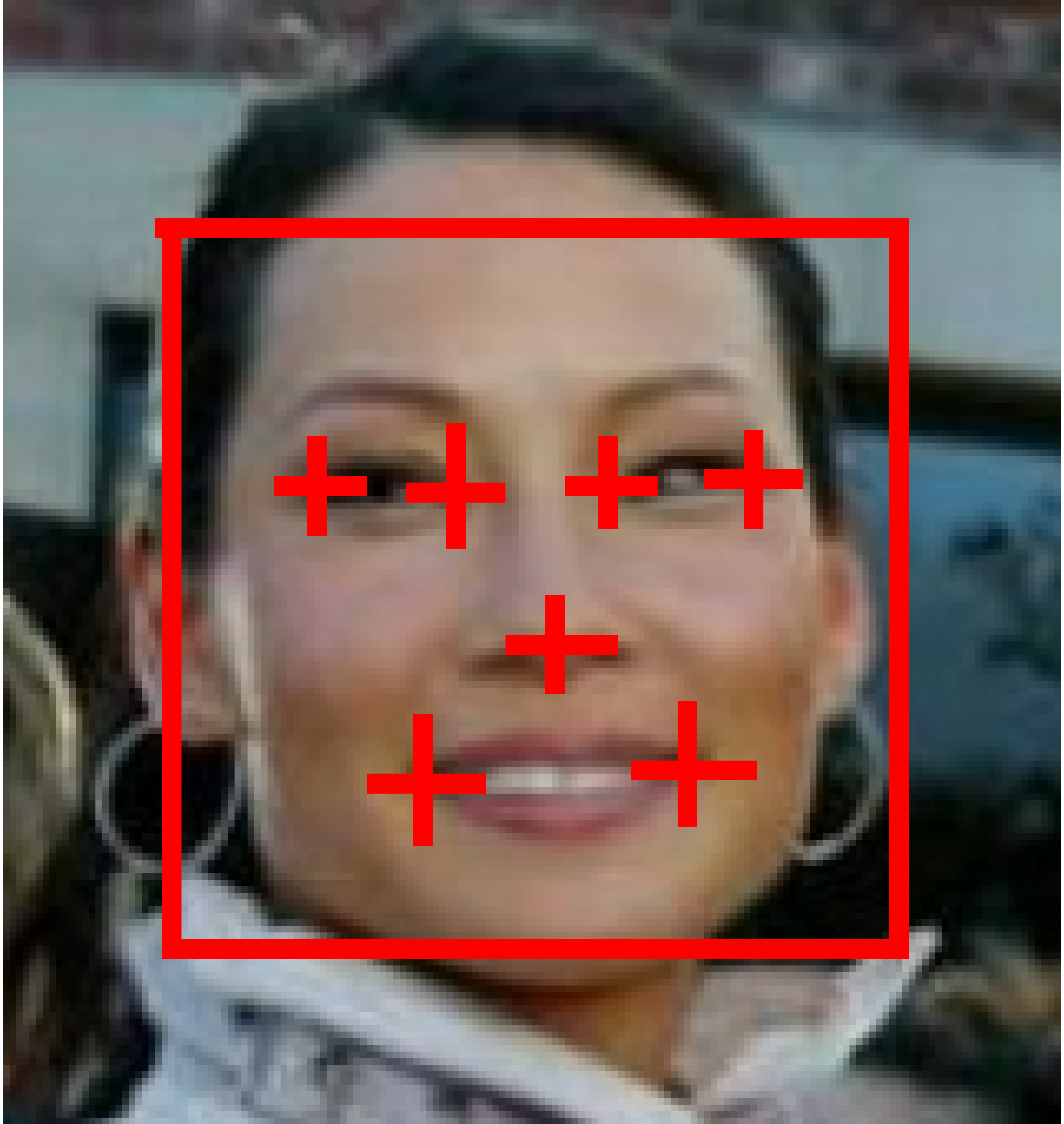
6
5
4
3
2
1
9
11
10
7
8
14
15
16
18
20
17
19
21

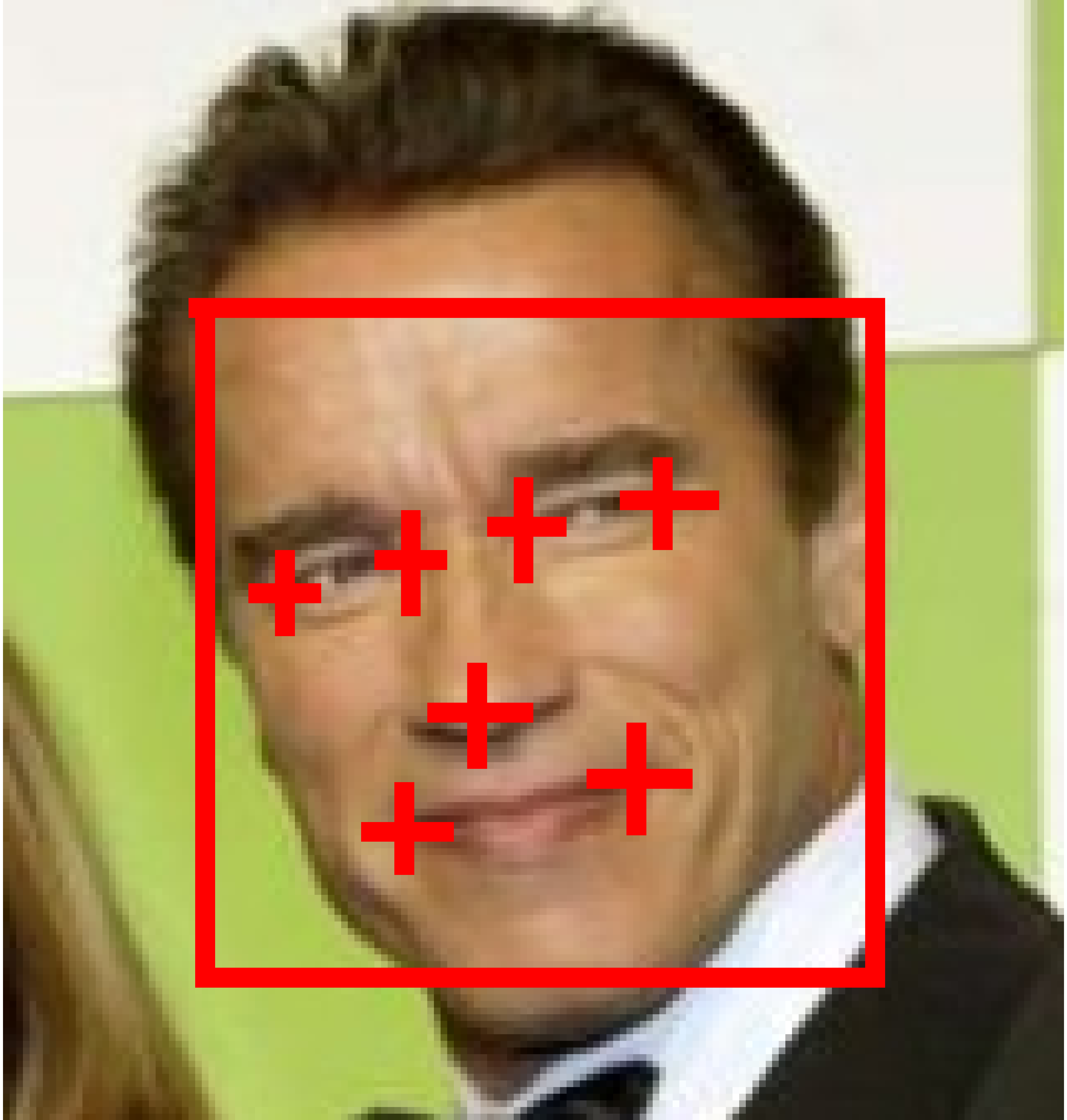


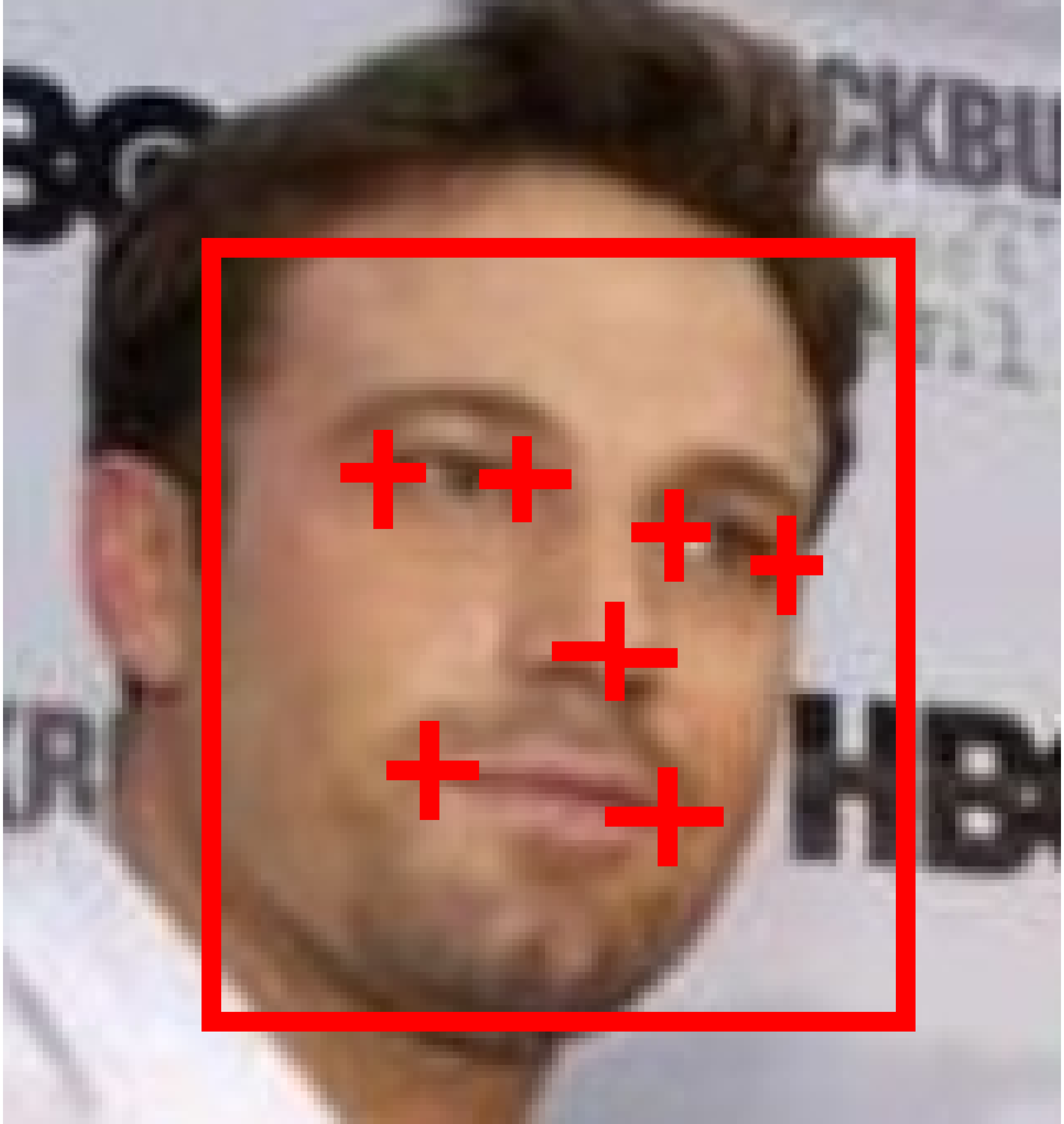
5
6
4
9
11
10
14
15
16
18
20
19
21

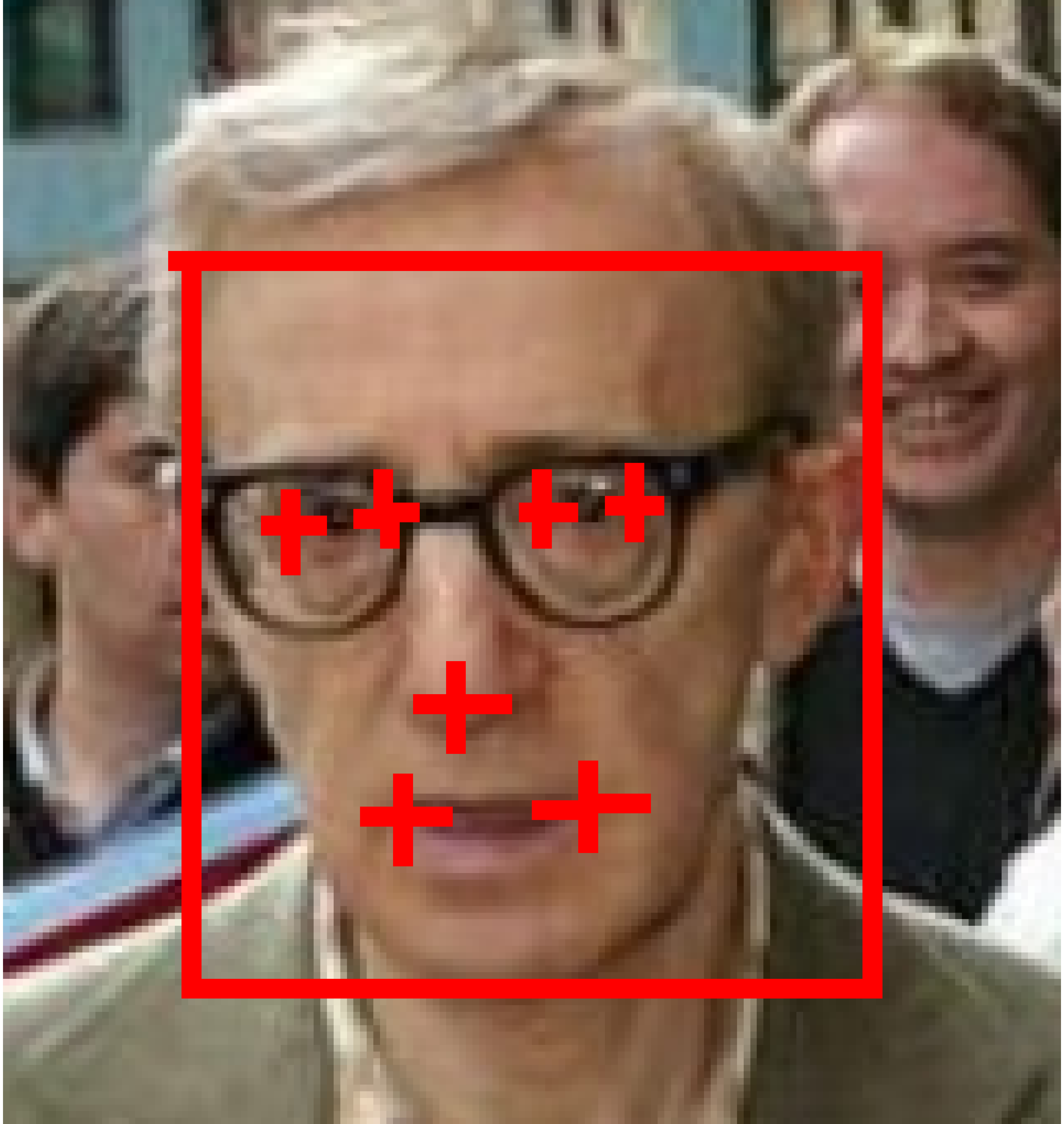

5
6
4
10
11
9
14
15
16
18
20
19
21



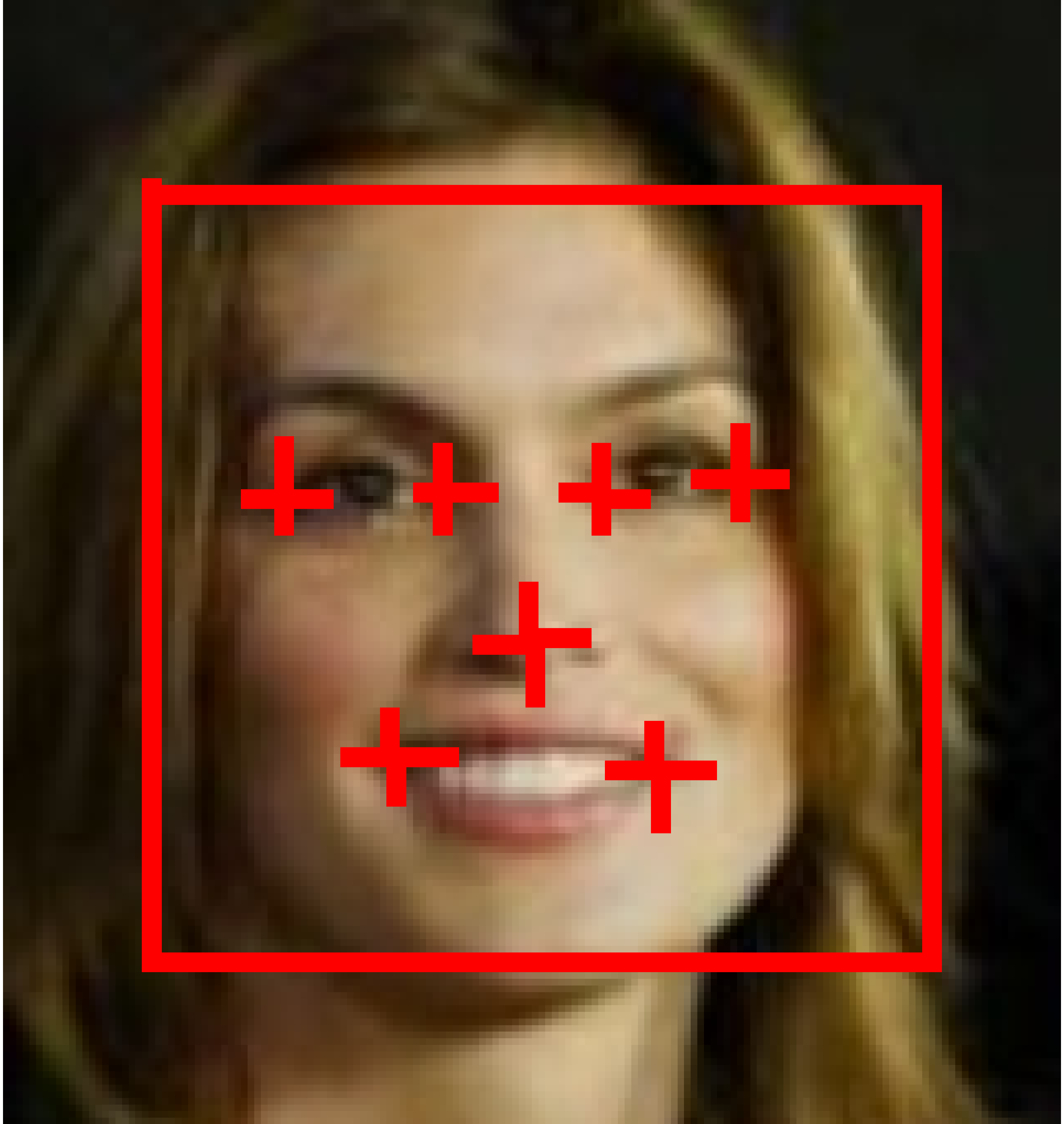


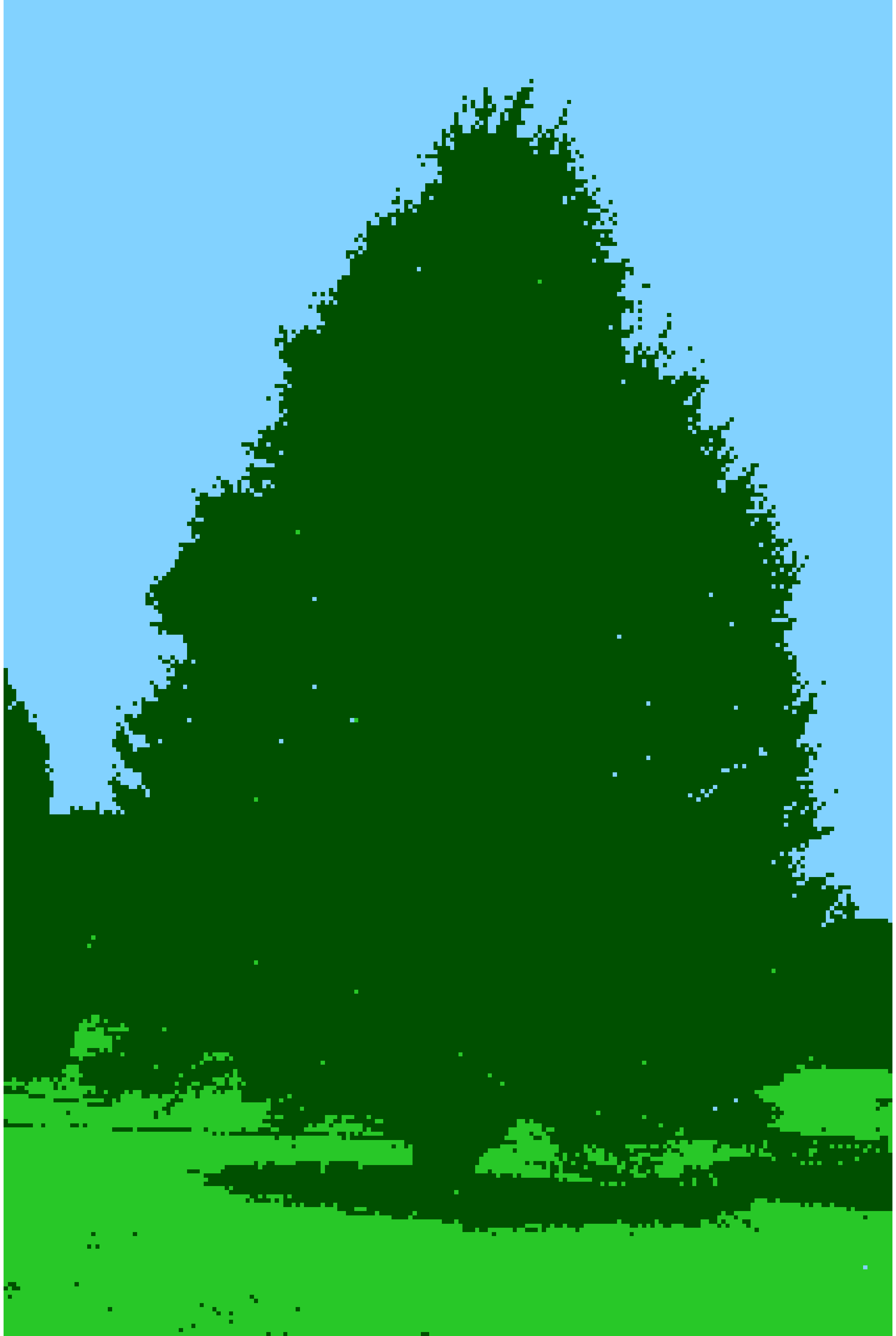


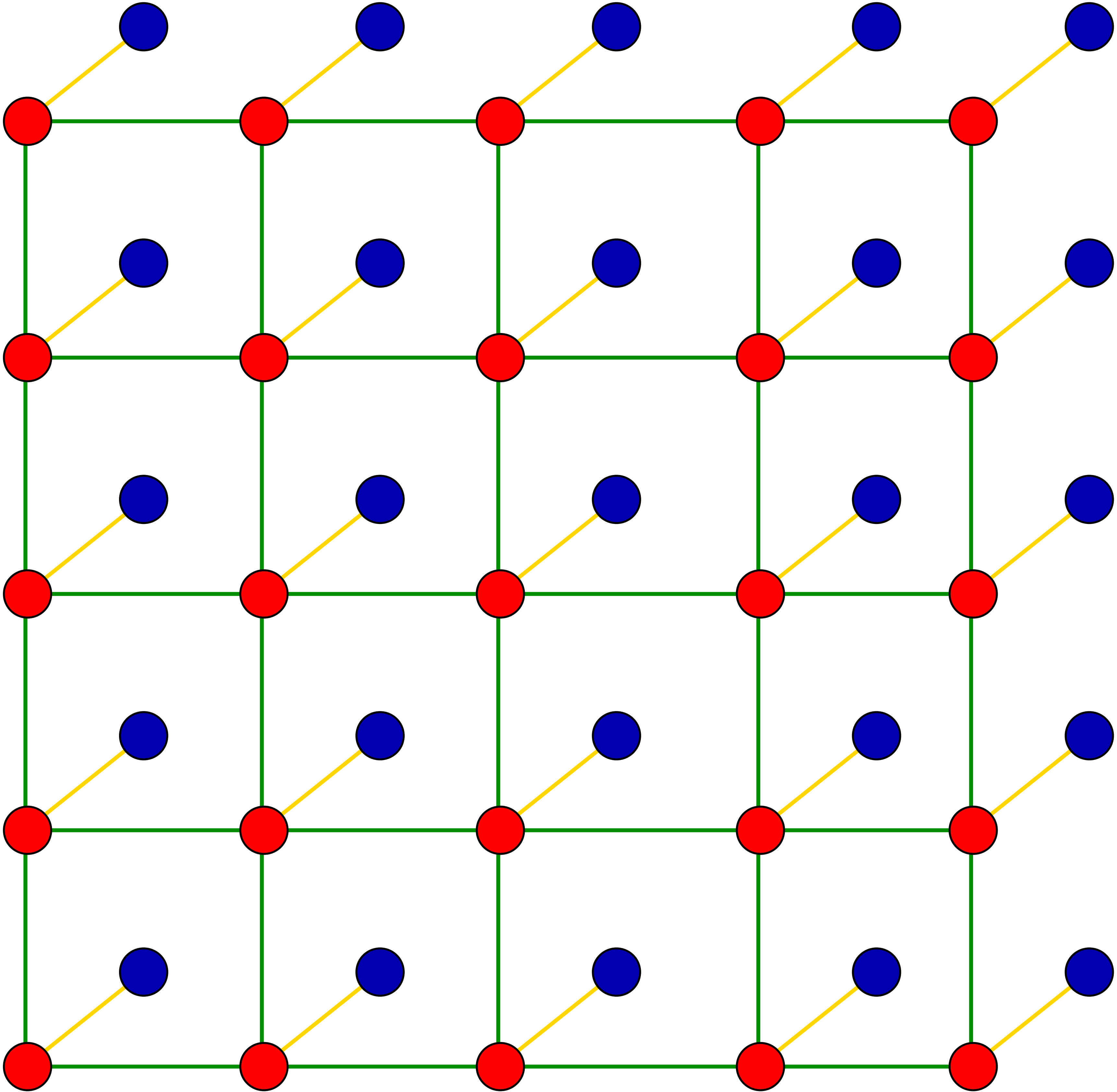


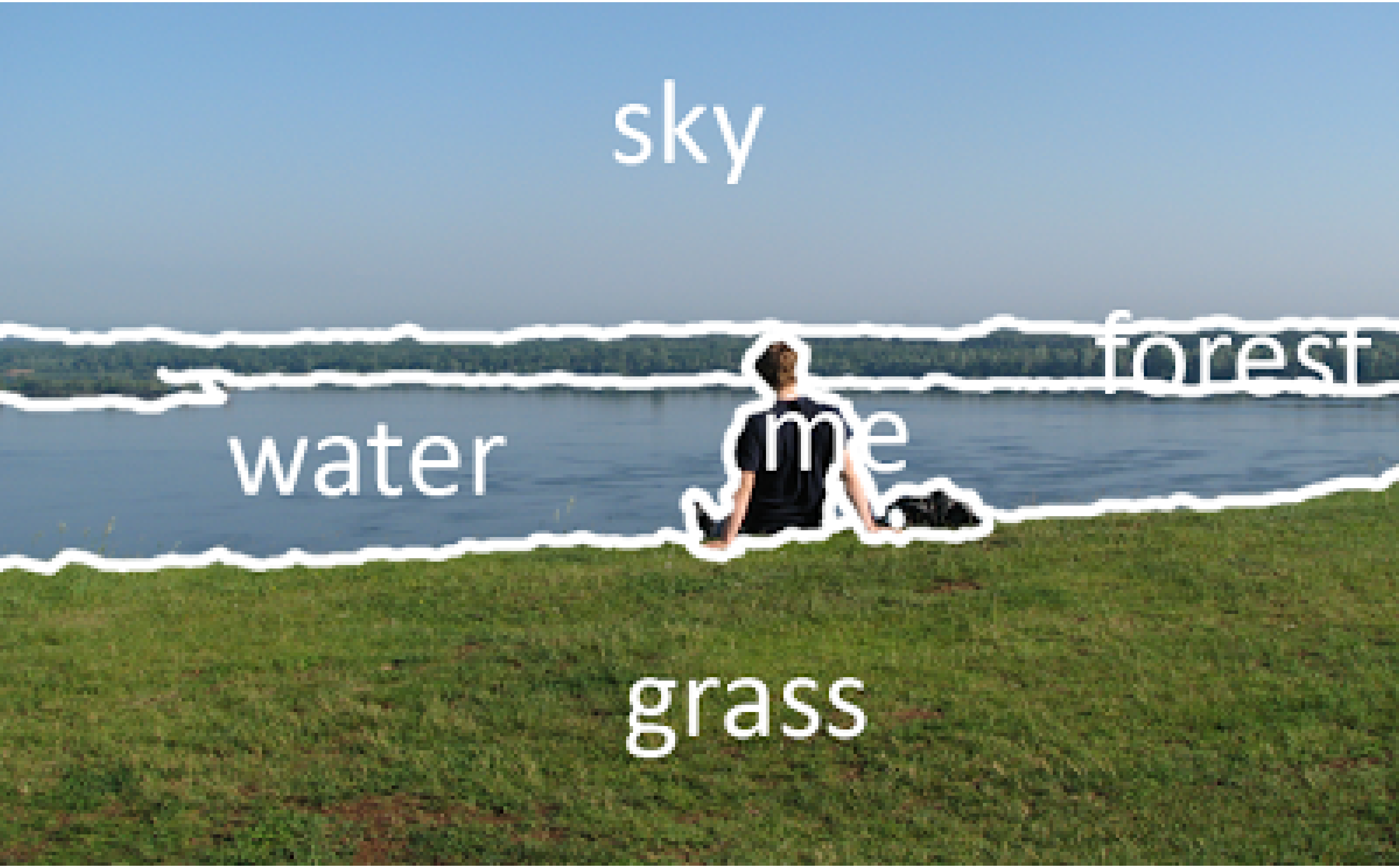


...








sky
forest
water
me
grass