Computer Vision II

Bjoern Andres

Machine Learning for Computer Vision
TU Dresden

April 27, 2020

Pixel classification

We consider:

- $n_0, n_1 \in \mathbb{N}$ called the height and width of a digital image, $V = [n_0] \times [n_1]$ called the set of pixels, and the grid graph $G = (V, E)$
- A non-empty set R whose elements are called colors
- A function $x\colon V \to R$ called a digital image

The task of pixel classification is concerned with making decisions at the pixels, e.g., decisions $y : V \to \{0, 1\}$ indicating whether a pixel $v \in V$ is of interest ($y_v = 1$) or not of interest ($y_v = 0$).

Pixel classification

Source: `https://www.pexels.com/photo/nature-flowers-garden-plant-67857/`

For instance, we may wish to map to 1 precisely those pixels of the above image that depict the yellow part of any of the flowers.

Pixel classification

We begin with a trivial mathematical abstraction of the task of pixel classification:

Definition. For any $c\colon V \to \mathbb{R}$, the instance of the **trivial pixel classification problem** w.r.t. c has the form

$$\min_{y \in \{0,1\}^V} \sum_{v \in V} c_v \, y_v \tag{1}$$

In practice, we would seek to construct the function c w.r.t. the image in such a way that

- $c_v < 0$ if we consider $y_v = 1$ the right decision
- $c_v > 0$ if we consider $y_v = 0$ the right decision

Pixel classification

Assuming the decision for a pixel $v \in V$ depends on the color $x_v \in R$ of that pixel only, we can

- construct a function $\xi \colon R \to \mathbb{R}$
- define $c_v = \xi(x_v)$ for any $v \in V$.

In some practical applications, e.g. photo editing, a suitable function ξ can be constructed manually, typically with the help of carefully designed GUIs.

Pixel classification

Assuming the decision for a pixel $v \in V$ depends on the location v and on the colors of all pixels in a neighborhood $V_d(v) \subseteq V$ around v, e.g.

$$V_d(v) = \{w \in V \mid \|v - w\|_{\max} \leq d\} \ ,$$

we can

- construct, for any pixel v, a function $\xi_v \colon R^{V_d(v)} \to \mathbb{R}$ that assigns a real number $\xi_v(x')$ to any coloring $x' \colon V_d(v) \to R$ of the d-neighborhood of v
- define $c_v = \xi(x_{V_d(v)})$ for any $v \in V$.

The task of constructing such functions ξ_v is typically addressed by means of **machine learning**, e.g., logistic regression or a CNN.

Pixel classification

In practice, solutions to the trivial pixel classification problem can be improved by exploiting **prior knowledge** about feasible combinations of decisions.

Firstly, we consider prior knowledge saying that decisions at neighboring pixels $v, w \in V$ are more likely to be equal ($y_v = v_w$) than unequal ($y_v \neq y_w$).

Definition. For any $c \colon V \to \mathbb{R}$ and any $c' \colon E \to \mathbb{R}_0^+$, the instance of the **smooth pixel classification problem** w.r.t. c and c' has the form

$$\min_{y \in \{0,1\}^V} \quad \underbrace{\sum_{v \in V} c_v \, y_v + \sum_{\{v,w\} \in E} c'_{\{v,w\}} \, |y_v - y_w|}_{\varphi(y)} \tag{2}$$

Pixel classification

A naïve algorithm for this problem is local search with a transformation $T_v \colon \{0,1\}^V \to \{0,1\}^V$ that changes the decision for a single pixel, i.e., for any $y \colon V \to \{0,1\}$ and any $v, w \in V$:

$$T_v(y)(w) = \begin{cases} 1 - y_w & \text{if } w = v \\ y_w & \text{otherwise} \end{cases} \quad .$$

Initially, $y \colon V \to \{0,1\}$ and $W = V$
while $W \neq \emptyset$
    $W' := \emptyset$
    for each $v \in W$
        if $\varphi(T_v(y)) - \varphi(y) < 0$
            $y := T_v(y)$
            $W' := W' \cup \{w \in V \mid \{v,w\} \in E\}$
    $W := W'$

Pixel classification

Suggested self-study:

- Construct a function ξ (Slide 5) for the task and image shown on Slide 3; visualize the output of ξ.
- Implement the local search algorithm (Slide 8) for the smooth pixel classification problem (2) such that $\varphi(T_v(y)) - \varphi(y)$ is computed in constant time.
- Apply your implementation to $c_v = \xi(x_v)$ and various positive constants c'.
- Discuss your results and compare these to the solutions of the trivial pixel classification problem (1) that is solved by your implementation for $c' = 0$.

Advanced self-study:

- Generalize your implementation to operate on classifications $y \colon V \to \{0, 1, 2\}$.
- Use your implementation to separate also the white leaves of the flowers in the image shown on Slide 3.