

# 1 Úvod

Bude se pracovat v Matlabu, v moodle je skupina.

*Poznámka* (Úkoly počítačového vidění)

Detekovat, najít a určit věci (tváře, jestli se smějí, nádory, znaky jako znaky na SPZ, biometrika jako oko, tvář, podpis, budovy, lidi, auta, ...) na obrázku

*Poznámka* (Rozpoznávání objektu tradičním náhledem)

pixels -> určení feature (např. rozdělení oblastí podle tvarů, důležitých bodů, barev, umístění hledaného objektu) expertem -> učení klasifikátoru -> rozpoznávání

## Definice 1.1 (Feature vektor)

Univerzální převedení obrázku do důležitých věcí.

Měl by být invariantní (měl by být stejný při rotaci a škálování), diskriminační (dobře rozdělovat objekty), kompaktní (co nejmenší)

## Definice 1.2 (Rozpoznávání)

Feature vektory tvoří prostor, kde se algoritmus naučí najít hranici, která odděluje objekty, co jsou nějaké a co jsou jinaké.

*Poznámka* (Klasifikace může být za pomoci)

Statistiky – Bayesova teorie rozhodování

Pravidel – Rozhodovací strom

Metriky – Technika nejbližšího souseda, diskriminační analýza?, podpůrné vektorové stroje?

Biologické inspirace – Neuronové sítě

## Definice 1.3 (Učení s učitelem)

Na training setu víme správné odpovědi.

## Definice 1.4 (Naivní Bayesův klasifikátor)

Vychází z podmíněné pravděpodobnosti na základě věcí, co víme.

┌

*Například*

└ Rozeznávání falešného úsměvu. 91,3%.

### **Definice 1.5** (Rozhodovací stromy)

Pravidly určíme, kterou větví se vydáme. Výhodou je, že nepotřebujeme koncept vzdálenosti.

*Například*

Rozpoznávání, co se děje na videu (např. vražda). 70% - 100%.

### **Definice 1.6** (K nejbližších sousedů)

Podíváme se na nejbližší známé objekty a rozhodneme se podle nich.

*Například*

Čtení znaků. 99% čísla, 94% velká a 89% malá písmena.

### **Definice 1.7** (Lineární klasifikace)

Rozdělení prostoru nadrovinou. Zlepšením je tzv. podpůrné vektorové stroje? (support vector machines)

*Například*

Rozpoznávání lidí a věku. 66,9 - 80% lidí, 63,8 - 75,7% věk.

### **Definice 1.8** (Umělé neuronové sítě)

Sítě z neuronů, které jsou velmi jednoduše simulovány, viz moje maturitní práce. (Na GitHubu pod uživatelem JoHavel).

*Například*

Rozpoznávání tváře. 90%. (80% na portrétech.)

*Poznámka* (Hluboké učení)

pixels -> učení se včetně feature -> rozpoznávání

*Například*

AlexNet (top 5 error cca. 16%)

Každoročně se pořádá ILSVRC (Imagenet Large Scale Visual Recognition Challenge), kde už se dosáhlo méně než 4% chyby (152 vrstev NN)...

*Poznámka* (Kombinovaný přístup)

pixely  $\rightarrow$  featury nalezené NN  $\rightarrow$  trénování klasifikátoru  $\rightarrow$  rozpoznávání

Hluboké učení se ale zdá účinnější.

### **Definice 1.9** (Klasifikační pipeline)

Features  $\rightarrow$  Výběr featur, jejich normalizace, ...  $\rightarrow$  klasifikace (výběr klasifikátoru, trénování klasifikátoru a následná klasifikace)  $\rightarrow$  evaluace  $\rightarrow$  features (respektive výstup, pokud jsme spokojeni).

### **Definice 1.10** (Features)

Vlastnosti objektu, spojují nějakým způsobem podobné objekty. Musí být diskriminativní (pokud nejsou dostatečně diskriminativní, jakože často nejsou, dá se ještě hledat rozdělení s nejmenší chybou). Měly by být kompaktní (co nejmenší, protože s příliš featurami nelze ve stejném čase dostatečně naučit klasifikátor).

### **Definice 1.11** (Normalizace feature)

Aby nenastával problém např. s odlišnými jednotkami, nebo s různě naškálovanými featury, normalizuje se vydělením referenční hodnotou, respektive různými statistickými metodami, např. standardizací ( $\tilde{x}_i = \frac{x_i - \mu}{\sigma}$ ), nebo  $3\sigma$  škálováním ( $\tilde{x}_i = \frac{x_i - \mu}{3\sigma} + 1$ ).

*Poznámka*

Rozhodovací stromy (a náhodné lesy), naivní bayesova metoda atd. nepotřebují normalizaci.

### **Definice 1.12** (Výběr featur)

Některé featury mohou být totožné, některé zas redundantní.

Takže vybereme nějakou podmnožinu featur a vyzkoušíme. Nebo se naopak podíváme na jednotlivé, ohodnotíme je a vybereme  $K$  nejlepších. Další možnost je přidávat je po jedné a testovat je ne samotné, ale s již vybranými. Nebo můžeme začít se všemi a odstraňovat nejhorší (zase oběma způsoby, sekvenčním i jedнокrokovým  $K$ ). Existuje i kombinovaný, který udělá nejdříve jedno a pak druhé. Pak existují i genetické a další algoritmy.

Podle čeho měřit: konzistence (jestli shodné hodnoty jsou ve shodné třídě, viz vzorec v prezentaci), nezávislost na ostatních featurách (opak tzv. korelace) + korelace s třídami, množství informace ( $\mathcal{I} = -\log(P(A = a_i))$ ,  $E(\mathcal{I}) = -\sum P(A = a) \cdot \log_2(P(A = a))$ ), co nám dá, vzdálenost mezi třídami po použití dané featury, ...