

I-BIOM: Zadanie č.3

GENEROVANIE PRÍZNAKOV

Vo vybranom programovacom jazyku implementujte program, vygeneruje príznaky pre danú biometriku, popr. ich klasifikuje. Pokračujte v tímoch, vybraných biometrikách a databázach z prvého zadania.

Čas odovzdania je určený časom vloženia do AIS. Deadline pre získanie 8 bodov je **04.04.2019 o 14:59** pre cvičenie o 15:00 a **04.04.2019 o 16:59** pre cvičenie o 17:00. Každý týždeň omeškania je penalizovaný stratou dvoch bodov.

Ucho

V rámci zadania budete pracovať na jednom z troch spôsobov generovania príznakov (PCA, LBP al. HOG) a následne budete tieto príznaky klasifikovať pomocou jednoduchej neurónovej siete.

- Implementujte program, ktorý načíta obstrihnuté a zarovnané uši (z druhého zadania al. dodaných anotácií) a vygeneruje z každého ucha žiadaný typ príznakov.
 - *Lokálne binárne vzory* (vypracujte, ak je vaše AIS ID deliteľné tromi)- ucho je rozdelené na podokná, z každého je získaný popis textúry pomocou binárnych vzorov. Z množiny týchto vzorov sú vybraté tie *uniformné* (teda také, ktoré majú buď 0 al. 2 prechody medzi nulami a jednotkami), ktoré sú následne rozdelené do histogramov (pre LBP s 8 bodmi je to 58 intervalov, ale môžete niektoré spojiť). Tieto histogramy sú nová reprezentácia ucha. **4b**
 - *Histogram orientovaných gradientov* (vypracujte, ak po delení vášho AIS ID tromi dostanete ostane zvyšok 1) - ucho je rozdelená na podokná, v každom sú pomocou Sobelovho operátora vyhodnotené veľkosti a smery gradientov, ktoré sú následne podľa smerov rozdelené do histogramu s 8 intervalmi. Tieto histogramy sú nová reprezentácia tváre. **3b**
 - *Analýza hlavných komponentov* (vypracujte, ak po delení vášho AIS ID tromi dostanete ostane zvyšok 2) - z dát nájdite priemerné ucho μ , kovariančnú maticu \mathbf{C} a jej vlastné vektory $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ a vlastné čísla. Každé ucho \mathbf{x} sa dá následne formulovať ako:

$$\mathbf{x} = \mu + w_1\mathbf{u}_1 + w_2\mathbf{u}_2 + \dots + w_n\mathbf{u}_n. \quad (1)$$

Vektor $\{w_1, w_2, \dots, w_n\}$ je nová reprezentácia ucha (zmenšíte jeho veľkosť s ohľadom na vlastné čísla kovariančnej matice). **3b**

- Natrénujte nad získanými príznakmi MLP sieť (môžete ďalej okresať vaše vstupné dáta tak, aby nebola výstupná vrstva príliš veľká). Rozdeľte dáta na tréningové a testovacie (validačné). Vytvorte neurónovú sieť (architektúra je na vás). Natrénujte neurónovú sieť. Vyhodnoťte chybu na testovacích dátach. **3b**
 - **Pre študentov, ktorí pracujú s LBP príznakmi, je toto nepovinná úloha.** Implementujte toto tréningovanie s pomocou cross-validácie. **1b**
- Pripravte sa na ďalšie zadanie, v ktorom budete trénovať konvolučné siete. Keďže použitá databáza možno nemá dostatok obrazov na tréningovanie takejto siete, skúsite, či pomôže ich umelé dogenerovanie vzoriek. Vyberte si spôsob generovania (šum, rozmazanie, zmena jasu, kontrastu, primeraná deformácia a pod. - kreativite sa medze nekladú) a aspoň zdvojnásobte veľkosť vašej trénovanej množiny. **1b**

Nepovinné úlohy

- **Pre študentov, ktorí pracujú s PCA príznakmi:** Vyberte si jednu tvár z vašej množiny a vizualizujte vzorec 1 s $n = 7$ (to znamená - pôvodné ucho, priemerné ucho, použité hl. komponenty a rekonštruované ucho). **1b**
- **Pre študentov, ktorí pracujú s HOG príznakmi:** Vyberte si ucho z vašej množiny a vizualizujte ho pre aspoň 5 rôznych nastavení parametrov HOG metódy (veľkosť okna, kroku a pod). **1b**

Tvár

- Implementujte program, ktorý načíta Vaše príznaky získané v predošlom zadaní a ďalej zmenší ich dimenziu pomocou algoritmu *Vreca príznakov* (z ang. Bag of Words, niekedy aj Bag of visual words). **Okrem študentov s PCA** - tí musia svoje príznaky pregenerovať nanovo - tentokrát ich negenerujú pre celé obrazy, ale pre podokná, t.j. reprezentácia tváre sú PCA príznaky z jednotlivých okien. **3b**
- Vygenerujte tieto príznaky pre rôzne počty "vriec" (clusterov) v algoritme. **2b**
- Nad novozískanými príznakmi opäť natrénujte MLP **2b**.
- Porovnajme medzi sebou úspešnosti sietí s rozdielnymi vstupmi v závislosti od rozdielnych parametrov algoritmu BoW (z druhej podúlohy). **1b**

Nepovinné úlohy

- Implementujte BoW "sami" - t.j. s pomocou algoritmov na clustering, s ktorými ste sa stretli minulý semester. **1b**

Dúhovky

V rámci zadania budete kódovať vaše normalizované dúhovky do binarizovanej matice (len s prvkami $[0,1]$; resp. čierna/biela). Využívať na to budete Gáborove filtre.

- Implementujte program, ktorý načíta vaše normalizované dúhovky (masky v tomto zadaní nebudete potrebovať) a nad nimi konvoluuje pomocou Gáborových filtrov:
 - Vygenerujte aspoň 5 rôznych Gáborových filtrov (štandard je 40), meniť môžete veľkosť Gaussovskej obálky alebo parametre waveletov. **2b**
 - Aplikujte tieto filtre na vaše normalizované dúhovky. Stačí vám pracovať s reálnou časťou výstupu. **1b**
 - Binarizujte výstup z filtra thresholdingom (napr. pomocou fáz). **1b** Výstup zobrazte - originále, po filtrovaní, po binarizácii (v programe alebo v priečinku) **1b**
- Nájdite najlepší z vašich filtrov:
 - Vyberte náhodne 10 párov dúhoviek rovnakého človeka (*true pairs*) a 10 párov, kde každá patrí inej osobe (*impostor pairs*). Prehľadne zobrazte binarizované výsledky nad touto množinou pre každý váš filter (napr. venovať každému stranu v dokumentácii). Rozhodnite, ktorý je najlepší (na pohľad by si mali byť *true pairs* podobné a *impostor pairs* nepodobné). **3b**

Nepovinné úlohy

- Pracujte aj s imaginárnou časťou Gáborovho filtra. Výstup kódujte pomocou fáz. **1b**
- Berte do úvahy viacero filtrov naraz (tzv. Gáborova banka). Vygenerujte zo všetkých filtrov jednu binarizovanú maticu (napr. pomocou *local energy* alebo *mean amplitude*). **1b**

Poznámky, spresnenia, odkazy

- Zadanie má tri časti:

1. vytvorenie kódu a spracovanie vzoriek
2. napísanie dokumentácie
3. osobné odovzdanie na cvičení

Aby bolo zadanie považované za odovzdané je potreba spraviť každú časť. Ne-
podceňte dokumentáciu - je potreba sa v nej vyjadriť ku vstupom a výstupom
a každej podúlohe v zadaní (aspoň niekoľkými slovami - metódy, výsledky, ak sa to
hodí aj obrázky).

- Je potrebné, aby bolo zadanie pred cvičením, na ktorom sa chystáte odovzdávať,
nahraté v AIS v prislúchajúcom mieste odovzdania.
- Zadanie bude obodované na cvičení po prezentácii pred cvičiacim. **Pochopenie
použitých metód a funkcií sa chápe ako prirodzená súčasť zadania a
neschopnosť zodpovedať na otázky o týchto metódach je penalizované
stratou bodov z danej časti**, a to aj v prípade, že kód je funkčný a správny.
- Pri overení schopnosti klasifikácie príznakov je vhodnosť experimentov vo vašej réžii.
Např. ak máte pocit, že niektorá vzorka/subjekt by mala byť z databázy vyradená,
aby boli výsledky reprezentatívne pre čo najvšeobecnejšie riešenie, nebojte sa tak
urobiť.
- V akom formáte budete načítavať vstupy a ukladať výstupy je na vás, majte ale na
pamäti, že výstupy z zarovnaní budete potrebovať v ďalších zadaniach.
- Môže vám pomôcť:

- <https://www.learnopencv.com/histogram-of-oriented-gradients/>
- https://globaljournals.org/GJCST_Volume13/1-Face-Recognition-using-Local.pdf
- <http://laid.delanover.com/explanation-face-recognition-using-eigenfaces/>
- <http://answers.opencv.org/question/35740/getgaborkernel-and-complex-values/>
- <https://cvtuts.wordpress.com/2014/04/27/gabor-filters-a-practical-overview/>
- <https://docs.opencv.org/3.0-beta/modules/imgproc/doc/filtering.html>
- <https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0f>

- https://docs.opencv.org/2.4/modules/features2d/doc/object_categorization.html
- <https://www.cl.cam.ac.uk/~jgd1000/csvt.pdf>