



Vysoké učení technické v Brně
Brno University of Technology



Fakulta informačních technologií
Faculty of Information Technology

Detekce krevních sraženin ve snímku sítnice oka

BIO - Biometrické systémy

Autor práce

Ladislav Šulák

Login

xsulak04

Brno 11/2016

Úvod

Cieľom bolo vytvoriť riešenie, ktoré bude detekovať krvné zrazeniny v snímkoch sietnice oka a graficky ich vyznačovať. V tejto práci bola ďalej požiadavka zamerať sa na hemorágie a aneurizmy. Čo sa týka snímkov takýchto sietnic, použili sme voľne dostupné databázy obsahujúce zdravé aj choré sietnice získané najmä od diabetikov v rôznych štádiách.

Práce, ktoré sa touto problematikou zaoberajú už existujú a vychádzajú väčšinou z predspracovania daného snímku sietnice napríklad pomocou rôznych metód pracujúcich s blízkosťami pixelov alebo morfológickými transformáciami¹ a na výsledok následne aplikujú nejaký známy algoritmus na strojové učenie s učiteľom pre zníženie nežiadúcich nepravdivých kandidátov (*false positives* a *false negatives*). Okrem tohto princípu je možné sa stretnúť ešte s takým, kde sa využívajú rôzne Gausovské filtre, alebo tzv. *Splat-based classification*², či Watershedov algoritmus³ na čo sa tiež aplikuje nejaká metóda strojového učenia, ako napríklad *Support Vector Machines*.

Dátová sada

Databáza *MESSIDOR*⁴, ktorá bola v tejto práci použitá, je voľne dostupná a obsahuje okolo 1200 záznamov sietnic oka získaných prevažne od diabetikov v rôznych štádiách a s rôznymi diagnózami. Nie všetky diagnózy sú pre našu prácu dôležité. Zaujímajú nás iba také, pri ktorých sú prítomnosti krvných zrazenín typu hemorrágie alebo aneurizmy. Táto databáza obsahuje snímky ako zdravých jedincov, tak aj jedincov, ktoré majú nasledovné diagnózy (pre presnosť ich nebudeme prekladať):

- *Retinopathy grade* - podľa stupnice *Keith-Wagner-Barker*⁵ má až 4 štádiá, pričom nás zaujímajú posledné dve.
- *Macular edema* - typicky sa pri nej netvoria krvné zrazeniny. Túto diagnózu nebudeme v našej práci ďalej uvažovať.

Okrem tejto databázy existuje ešte niekoľko ďalších, ktoré sú voľne k dispozícii a obsahujú pre nás relevantné dáta. Napríklad *DiaRetDB1*⁶ alebo *STARE*⁷ (*Structured Analysis of the Retina*), ktorá obsahuje mnoho rôznych diagnóz a okolo 400 snímkov sietnic. Pre našu prácu by boli dôležité najmä kategórie diagnóz 7, 9 a 11 pri vyšších štádiách. Databáza *MESSIDOR* bola ale najrozsiahlejšia a pre prípady našej práce dostačujúca.

Návrh a implementácia riešenia

Ako implementačný jazyk bol zvolený Python s verziou interpretu 3.5.2. To bolo rozhodnuté kvôli tomu, že je v ňom vývoj rýchly a obsahuje mnoho knižníc, ktoré obsahujú už hotové moduly pre predspracovanie obrazu alebo rôzne algoritmy strojového učenia. Je pravda,

¹<http://ir.uiowa.edu/cgi/viewcontent.cgi?article=1015&context=omia>

²<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6352921&tag=1>

³<http://www.rroij.com/open-access/detection-of-retinal-hemorrhage-in-fundusimages-by-classifying-the-splat-featuresusing-svm.pdf>

⁴<http://www.ias-iss.org/ojs/IAS/article/view/1155>

⁵<http://ophthaclassification.altervista.org/hypertensive-retinopathy-classification-keith-wagner-barker>

⁶http://www.it.lut.fi/project/imageret/diaRetdb1_v2_1

⁷<http://cecas.clemson.edu/~ahoover/stare>

že je to interpretovaný jazyk, avšak mnoho modulov je kompilovaných, čo má za následok často vysokú rýchlosť vykonania výsledného programu. Jadro nášho riešenia sa opiera o knižnicu OpenCV.

Naše riešenie je založené na objektovo orientovanom prístupe, pričom boli implementované dve triedy:

- *DBextractor*, ktorý v súborovom systéme nájde všetky vstupné snímky sietnice oka uložené ako obrázky. Spolu s nimi aj informácie o týchto snímkoch, teda konkrétne diagnózy a ich stupeň. Tieto informácie sa predajú nasledujúcej triede ako zoznam obsahujúci dvojice. Prvý prvok v dvojici je meno vstupného súboru a druhý označuje, či sa v ňom krvná zrazenina nachádza alebo nenachádza.
- *RetinaProcessor*, prijímajúci zoznam vstupných dát. Každý obrázok zo vstupu spracuje pomocou prahovania a rôznych morfologických transformácií a úprav. Následne sa pokúsi označiť miesta, ktoré by mohli predstavovať krvné zrazeniny. Na konci spracovania každého snímku uloží nový snímok do súborového systému spolu s informáciou o počte krvných zrazenín, ktoré sa našli. Takáto množina výstupných dát sa ukladá podobne ako je uložená vstupná databáza. Na konci spracúvania sa určí presnosť detekcie, no žiaľ vzhľadom na to, že sme nemali informáciu o počte krvných zrazenín vo vstupných snímkach, presnosť vyhodnocujeme iba na základe toho, či sa nejaká zrazenina našla a mala tam byť alebo nie.

Princíp detekcie krvných zrazenín

V prvom kroku dochádza ku transformácii obrazu do šedotónovej varianty (obr. 1a). Následne sa oddelí sietnica od okolia (obr. 1b a 1c) a zvýraznia sa tmavé oblasti (obr. 1d). V prípade, že boli nejaké tmavšie objekty pri okraji sietnici, na záver sa zlúčili s tmavým pozadím a vznikol omnoho väčší objekt. Kvôli tomu sme označili pozadie sietnice na bielo. V tomto kroku máme teda vo výsledku krvné zrazeniny, žily a žltú škvrnu (*Macula*). Postupným aplikovaním morfologických metód získavame zvýraznenú žltú škvrnu a krvné zrazeniny (obr. 1e a 1f a 1g). Čo sa týka žíl, v mnohých prípadoch sme ich už v tomto kroku správne odfiltrovali, no úspešnosť závisí od diagnózy pacienta, pretože aj tie môžu mať rôznu hrúbku. Pri morfologických úpravách sme zvolili nasledovné:

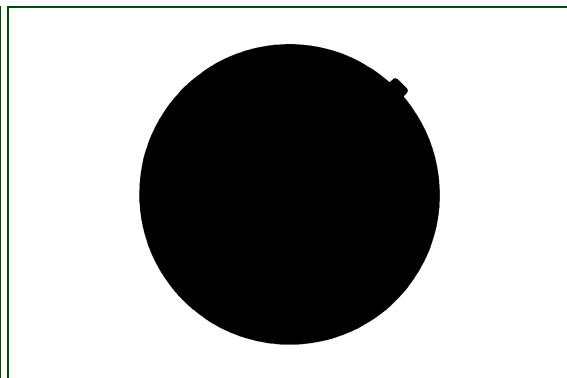
- operáciu dilate s nastavením kernelu o veľkosti 2x2 pixely s jednou iteráciou, čo má spolu s ďalšou operáciou za následok redukovanie šumu a necelistvých oblastí ako sú zostatky po žilách,
- operáciu close s rovnakým kernelom ako predchádzajúca operácia, no s počtom iterácií rovných 3,
- operáciu open s kernelom 5x5 pixelmi a s 3 iteráciami. To bolo zvolené kvôli tomu, aby sa vyplnili oblasti s medzerami, ktoré odfiltrovali predchádzajúce 2 metódy. Miesta, ktoré sa vyplnia sú tie, ktoré obsahujú naokolo tmavšie pixely, čo znamená, že sa jedná o nejaký celistvý objekt.

Následne sme pomocou knižnici OpenCV použili triedu *SimpleBlobDetector* ktorá dokáže nájsť a farebne vyznačiť objekty podľa nastavení rôznych filtrov (obr. 1h), v našej práci sme ich použili k ďalšiemu odfiltrovaní žíl, jedná sa o:

- úroveň zaoblenia objektu, ktorá vychádza z toho, že krv sa rozlieva a tečie, nemôže byť teda príliš hranatá,



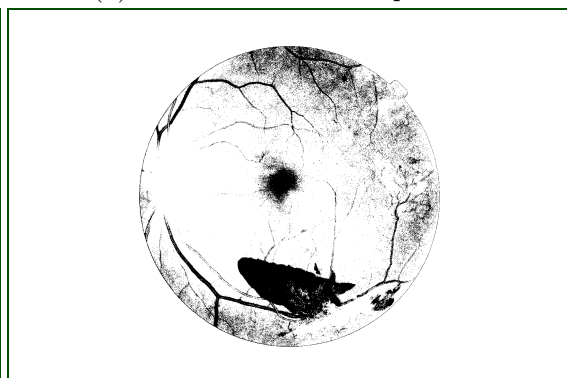
(a) Šedotónový obrázok.



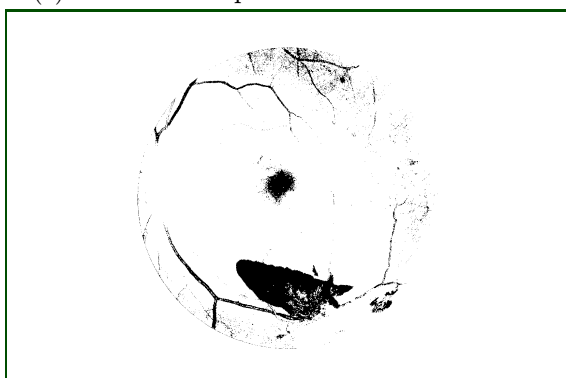
(b) Oddelenie sietnice od pozadia.



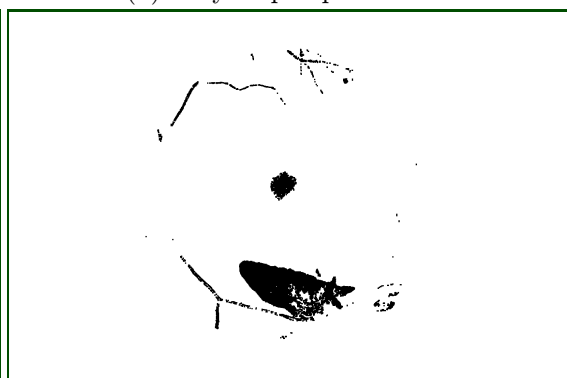
(c) Odfiltrovanie pozadia a získanie sietnice.



(d) Prvý stupeň prahovania.



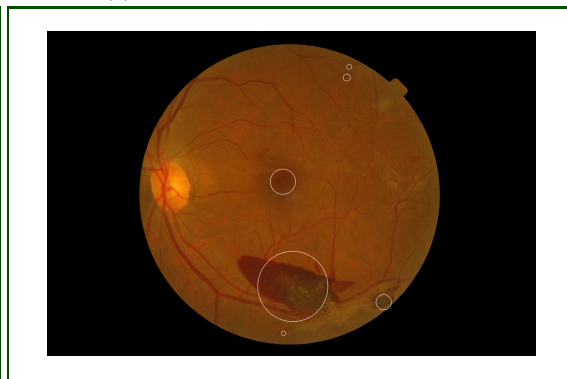
(e) Morfológická metóda *dilation*.



(f) Morfológická metóda *closing*.



(g) Morfológická metóda *opening*.



(h) Zvýraznené oblasti v pôvodnom obrázku.

- úroveň konvexity, vychádzajúca z podobnej vlastnosti krvi ako v predchádzajúcom bode,
- úroveň inercie, ktorá odfiltruje prílišne elipsovité útvary, teda mnoho útvarov reprezentujúcich žily. Tento filter je veľmi jemný, pretože krv môže mať aj takúto podobu,
- minimálna veľkosť objektu musí byť stanovená kvôli tomu, že veľmi malé objekty sú typicky pozostatky po žilách.

Potrebné knižnice a použitie

Pre úspešné používanie je potrebné mať nainštalované:

- Python 3 a knižnicu *xlrd*⁸, dostupnú aj z tzv. *Python Package Index* s verziou 3 (*pip3*)
- OpenCV⁹ pre Python 3¹⁰ (verzia použitá v tejto práci bola 3.1.0)

Riešenie sa má a predpokladá nasledovnú štruktúru:

- *DB/././* - teda zložku DB obsahujúcu rôzne databázy, v našom prípade iba *adcis*, ktorá ďalej obsahuje datasety pre rôzne druhy meraní, prípadne rokov. Tie už potom obsahujú obrázky zachycujúce sietnice oka. Medzi nimi sa nachádzajú aj zdravé aj poškodené sietnice, informácie o každom snímku sa nachádzajú v dokumente vo formáte xls.
- *retina_processing.py* - skript v jazyku Python, ktorý získa dáta z databázy, spracuje každú snímku sietnice a výsledok uloží do ďalšej databázy do súborového systému.
- *README* - manuál na prípravu prostredia pre naše riešenie. Obsahuje predovšetkým postup ako nainštalovať knižnicu OpenCV a všetky potrebné závislosti.
- *xsulak04.pdf* - táto dokumentácia.

Pri behu programu *retina_processing.py* dochádza k priebežnému ukladaniu výsledkov do zložky *ResultDB/*, ktorá má rovnakú štruktúru ako vstupná databáza, akurát informácie o výsledných snímkoch a krvných zrazeninách sú uložené v súboroch s názvom *hemorrhageStats.csv*.

Záver

Riešenie, ktoré sa podarilo v tejto práci navrhnuť a implementovať dokáže detekovať a zvýrazniť krvné zrazeniny na sietnici oka. Nemáme presnú informáciu o tom, koľko krvných zrazenín sa vo vstupných obrázkoch nachádza, máme iba informáciu o tom, či sa tam nachádza aspon jedna, alebo žiadna. Taktiež sa nám nepodarilo odfiltrovať slepú švrnu, ktorá má podobnú signatúru ako krvné zrazeniny. Z týchto dôvodov nie je možné presne určiť kvalitu a presnosť nášho riešenia, no ak by sme porovnávali informáciu, že sme našli aspon 1 zrazeninu a naozaj sa tam vyskytuje, tak je to nad databázou 1200 sietnic približne 50%.

⁸<http://pythonhosted.org/pyserial>

⁹<http://docs.opencv.org>

¹⁰<https://pypi.python.org/pypi/opencv-python/3.1.0>

Výstupom je databáza, ktorá má rovnakú štruktúru ako tá vstupná. Výsledné snímky sietnic navyše obsahujú označenia jednotlivých oblastí s krvnými zrazeninami a v každej podzložke danej databázy sa nachádza CSV súbor obsahujúci štatistiky o každom snímku a počte nájdených zrazenín.

Čo sa týka ďalšieho rozšírenia, okrem využitia snímok z ďalších voľne dostupných databáz by stálo za to použiť ešte nejaké iné metódy pre spracovanie obrazu, ktoré ponúka knižnica OpenCV, prípadne aj mimo nej. Taktiež nebol použitý algoritmus na učenie s učiteľom, ako napríklad Support Vector Machines, alebo ďalšie typu Naive Bayes, ktoré podľa literatúry produkujú oveľa presnejšie výsledky. Taktiež je dôležité dokázať rozlíšiť slepú škvrnu od krvnej zrazeniny väčšej veľkosti, čo sa v tejto práci nepodarilo.