

Tibor Milić: Prepoznavanje vremena promjene prikaznice u video snimci predavanja

Dnevnik rada

Kratki sadržaj:

1. Priprema za rješavanje problema
2. Upoznavanje s problemom i usmjeravanje na određene teme
3. Rješavanje problema i testiranje
4. Proba 01, Proba 02, Proba 03
5. Zaključak - Analiza kvalitete rada, dodaci i komentari za poboljšanje

1. tjedan (14.03.2016 - 20.03.2016)

Dosadašnji rad:

Instalacije.

Ubuntu 12.0.4 LTS

- Python Interpreter 2.7 & 3.4.4.
- PyCharm IDE
- Text-Edit, Editors
- Downloaded: avconv
- Functions for Python: sciPy and some other

All ready and installed.

Proučavanje ulaznih dokumenata:

1. Skinute dvije snimke predavanja
 - With Mozilla Firefox Addons
2. Naredba za splitanje videa u seriju slika - Nije radila u danom obliku, čitanje dokumentacije za avconv
 - Literatura:
<http://www.tecmint.com/avconv-command-examples/>
<http://www.dototot.com/how-to-trim-videos-with-the-command-line-using-avconv/>
 - Zaključak: Spremljene i komentirane naredbe za korištenje avconv alata, te isprobavanje mogućnosti/funkcionalnosti
3. Proučavanje skripte za usporedbu slika:
 - Rad sa slikama: SciPy, čitanje dokumentacije, te razumijevanje rada sa slikama (2D i 3D arrays), funkcije sum, average, norms... i alternative
 - Literatura:

<http://docs.scipy.org/doc/>

<http://www.pyimagesearch.com/2014/09/15/python-compare-two-images/>

<http://effbot.org/imagingbook/image.htm>

- Zaključak: Puno dostupne literature, odlučiti se za jednu seriju biblioteka i naučiti raditi sa slikama kroz dane funkcije.

Učenje Python-a i XML-a:

1. Video Tutorials

- Python programming lesson 43 min - Whole language syntax, Videos from Coursera (6 hours of learning and watching)
Literatura: <https://www.youtube.com/watch?v=N4mEzFDjqtA>
- <https://www.youtube.com/watch?v=tAN-1xUsftg&list=PLBB413675AFBDC1F4>

2. Coursera Tutorials

- Coursera materijali - prezentacije i knjiga o Pythonu (free)
- Literatura: <https://www.coursera.org/learn/interactive-python-1>

3. Other Webpages

- <https://wiki.python.org/moin/MovingToPythonFromOtherLanguages>
- <http://www.tutorialspoint.com/python/>

Daljnji koraci

1. Istražiti funkcije za obradu slika, te OCR
2. Obraditi dana videopredavanja tj. splitati ih u slike.
3. Izračunati norme
4. Usporediti par slika i vidjeti kako to sve funkcionira
5. Naučiti što bolje sam jezik Python

2. tjedan (21.03.2016 - 27.03.2016)

Dosadašnji rad

1. Istraživanje o funkcijama za obradu slika
 - Literatura:
 - http://www.scipy-lectures.org/advanced/image_processing/#opening-and-writing-to-image-files
 - <http://jehiah.cz/a/creating-images-with-numpy>
2. Splitanje 10-minutnog videa u 1366×768 jpg slike. 602 slike 1FPS
 - Zaključak možda je i previše 1-FPS, jer postoji barem 300 plavih slika. A prezentacija se ne mijenja barem 5-10 ili više sekundi. Ali ostao bi pri 1FPS.
 - Odluka: Slike ostaju u rezoluciji iz koje dolaze (videopredavanje), 1 FPS
3. Učitao dvije početne slike izračunali se n_{norm} i Manhattan norm.
4. Učitao dvije plave slike - norm-ovi su 0.0, dakle usporedba radi
5. Učitao plavu sliku i sliku u powerpointu, normovi su različiti, nisu 0.0!

Daljnji koraci

Učitavanje n slika u Python, pretpostavljeni oblik image-0000.jpg (različite mogućnosti za ekstenziju, ".jpg, .png")

1. Funkcije za OCR

- Napisati kod (već je 80% gotovo) za praćenje intervala plavih slika u slikama stvorenim iz probnih 10 minuta video predavanja
- Daljnje usporedbe, probe, te poboljšanje koda + još neke sitnice da bude u skladu s programom rada
- Proučiti rad s OCR-om, te implementirati algoritam za traženje powerpoint ključnih riječi. Te takvih intervala u snimci videopredavanja

2. Instalirati potrebno okruženje na windowsu (jednostavnije i brže izvođenje skripte za isprobavanje)

- AvConv
- Scipy - Anaconda Distribution

3. tjedan (28.03.2016 - 03.04.2016)

Dosadašnji rad

Primjeri za organizaciju popunjavanja izvještaja.

Primjer 01.

TO – DO

1. Instalacija Scipy i AvConv paketa na windowsu. STATUS: DONE
2. Instalacija OCR funkcija STATUS: None
3. Coursera Python Course. STATUS : First part DONE , Second Started (5 Hours of watching and learning)
4. Proučavanje OCR metoda. STATUS : 90% DONE
 - <https://saxenarajat99.wordpress.com/2014/10/04/optical-character-recognition-in-python/>
5. Export modul. Napisana funkcija koja prima listTransition i listTitle te u .csv dokument upisuje potrebne informacije nakon analiza videopredavanja.
6. Algoritam za praćenje pojavljivanja powerpoint sučelja, STATUS: 95% DONE
 - OCR metode koje koristim u slučaju čiste fullscreen prezentacije daju odlične rezultate točnosti do 99%.
 - Problem: Kod slike u slici, koja sadrži čudno znakovlje, program ne radi - ne može pročitati te znakove koje je OCR napravio.
 - Rješenje: dokumenti se otvaraju u encoding = utf-8. Nakon toga sve daljnje je radilo
 - OCR metode sam koristio na Video Snimci01, koja ima slabiju rezoluciju (ne 1024×768) te je

time manje kvalitetna slika, a tako i sve slike koje sam radio.

- Problem: Nemoguće je u mutnoj slici, sa sitnim slovima kao što je sučelje powerpointa prepoznati sa OCR tražene vrijednosti (home, file, slideshow, powerpoint itd)
- Rješenje: avconv nudi smiješne rezolucije npr. 1920×1080, nakon takve izvedbe OCR uspjeva pronaći 90% onoga što se traži, i na temelju toga radi. Također nudi i izradu kvalitetnijih slika ali pri ostajanju na početnoj rezoluciji (1024×768), igranjem s takvim opcijama daje normalne rezultate.

7. U daljnjem razmatranju uzimati te kvalitetnije uzorke, i proučavati OCR funkcije u detalje, te pronaći dobar algoritam za praćenje tema u takvim uvjetima. **STATUS: Working**

- Problem: Što se smatra temom? Drugačiji naslov, velike pod teme? Koliko tema u snimci od 2 sata smije biti? Itd...

8. Prebaciti dosadašnje testiranje na linux, (razlika je samo u tome da na linuxu postoje još neke dodatne mogućnosti i funkcije koje ću tako i istražiti) **STATUS: DONE**

9. Isplati li se brisati iste slike. **STATUS : Working**

- Analizu provodim pomoću norma, ako su slike mrvicu različite to ne uspjeva, zatim pomoću OCR-a, ako se slike razlikuju u manje od 5 linija (proizvoljna konstanta) tretiram ih kao jednake, Brisanjem istih slika u jednom prolazu, te kasnije uspoređivanje manjeg broja slika, pomoću istih algoritama nema smisla. Ali je dobra referenca, odnosno ako iz analize izbacimo OCR, slike se izbrišu u daleko manjem vremenu negoli je potrebno za jedan OCR. Time bi se dobilo na vremenu.

10. User input modul. **STATUS : 90% DONE**

- Omogućava ručno unošenje radnog direktorija, naredbe s parametrima za avocnv, te OCR naredbe...Proizvoljan izbor broja slika koje korisnik želi obraditi i drugo.

Primjer 02.

Proba 01:

Slika je nastala iz skripte koja prolazi kroz n slika te vraća vremena pojavljivanja plavih slika (blue-image) za proizvoljno izmješane slike (tako da se provjeri ispravnost koda).

1. Praćenje plave slike u seriji slika

<http://s12.postimg.org/kron3uaml/Proba01.png>

Daljnji koraci

1. Proučavanje biblioteka i novih rješenja za **OCR STATUS: Started**
2. Osmišljavanje algoritma za user input, te algoritma za OCR slike po dijelovima, proučavanje dostupne literature... **STATUS: Started**
3. **Pass second and third** Coursera Python Course. **STATUS : In progress**

4. Proučavanje Tkinter modula za izradu GUI aplikacije **STATUS** : 20% Done (**not important at this moment**)

5. Optimization of written algorithms. **STATUS: In progress**

4. tjedan (04.04.2016 - 10.04.2016)

Dosadašnji rad

Proba 02:

1) Uspješan prolazak kroz 902 slike

2) Formati: 1920×1080 (dobra kvaliteta, nije ni bitno jer se kasnije brišu slike, a odlično može prepoznati OCR sve na takvim slikama)

3) Output datoteka (doduše previše "here goes slide" umetanja...jer još nije napisan algoritam):

Odlično praćenje plave slike, malo lošije za powerpoint:

Razlog: Sitno zamućivanje određenih slika radi brzog prebacivanja sučelja powerpointa, pa je odstupanje za 5-10 frame-ova (tj. 5-10 sec)

Rješenje: Postaviti algoritam na jednostavnije linijsko pretraživanje i smanjiti normu, te isprobati ponovo samo na dijelu gdje se nalazi powerpoint.

Izveštaj: **Works perfectly** , od početka do kraja, od frame-a do frame-a, našao powerpoint sučelje i nakon što ga više nije bilo zaustavio se.

4) Ispravljanje sitnih grešaka, i omogućavanje iznimaka za pojedine procese u algoritmu

5. tjedan (11.04.2016 - 17.04.2016)

1. Proučavanje biblioteka i novih rješenja za **OCR STATUS: In progress**

2. Osmišljavanje algoritma za user input, te algoritma za OCR slike po dijelovima, proučavanje dostupne literature... **STATUS: Working**

3. **Pass second and third** Coursera Python Course. **STATUS** : **Second DONE!, Third Started**

4. Proučavanje Tkinter modula za izradu GUI aplikacije **STATUS** : 20% Done (**not important at this moment**)

5. Optimization of written algorithms. **STATUS: 95% Done**

Proučavanje literature:

https://en.wikipedia.org/wiki/Canny_edge_detector

http://scikit-image.org/docs/dev/auto_examples/applications/plot_rank_filters.html

<http://stackoverflow.com/questions/17103735/difference-between-edge-detection-and-image-contours>

https://en.wikipedia.org/wiki/Minimum_bounding_box

https://en.wikipedia.org/wiki/Precision_and_recall

https://en.wikipedia.org/wiki/F1_score

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>

<http://www.danvk.org/2015/01/09/extracting-text-from-an-image-using-ocropus.html>

http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html

https://en.wikipedia.org/wiki/Artificial_neural_network

https://en.wikipedia.org/wiki/Long_short-term_memory

<https://en.wikipedia.org/wiki/HOCR>

<http://www.danvk.org/2015/01/11/training-an-ocropus-ocr-model.html>

<https://github.com/tmbdev/ocropy/issues/16>

https://en.wikipedia.org/wiki/Supervised_learning

I još dosta odličnih prezentacija i stvari za proučiti, ali to ću organizirati kada izdvojim ono najbitnije i najkorisnije od svega, pa staviti ovdje.

Planovi za sljedeći tjedan:

1. Implementirati osnovnu funkcionalnost obrade slide-a, rasčlamba algoritma na User Input (template slide-a), obrada slike u odnosu na taj Template te izvođenje potrebnih OCR funkcija i analiza teksta za određivanje teme, broja slide-a, eventualnog sadržaja i sl.
2. Isprobati razne mogućnosti, u vezi kvalitete algoritma, kvalitete slika i videa, isprobati 2-3 template-a (različitih slide-ova) koji se većinom pojavljuju, te uspoređivati razne outpute: 1, 2, 5, 10 fps-a, razne kvalitete i rezolucije slika i sl.
3. Ukomponirati sve navedeno i napraviti Probu 03.

6. tjedan (18.04.2016 - 24.04.2016)

Započeo sam točke pod 1 i 2, **ali pošto** ć e sada ispiti bit će to malo sporije, dati ću sve od sebe da napravim što više.

Što se tiče dogovora na zadnjem sastanku, sada koristim “.png” formate, te je slika od 1024 x 768 dovoljno dobra, no možda ne i savršena, dolazi još uvijek do nekog odstupanja, tako da je OCR točan u 98% slučajeva.

Za potpunu točnost format 1920 x1080 (slike koja nam ionak nije važna) bio bi sigurno 100% točan i pogodan za OCR.

Kada dovršim pisanje algoritma i sve ukupno isprobam na 3-4 videa, vidjeti ćemo koji nam rezultati više odgovaraju.

Mogućnost definiranja predložka slajda je u izradi. **A ostale sitnice** kao što je “export početaka određenih djelova predavanja” a ne “kraj” su sređene i rade odlično.

Napisao jednostavan alogoritam po jednostavnom predložku:

Zaključak: OCR radi brže u ovisnosti o veličini slike, manje od pola sekunde po djelu.

Uspješno exportanje Naslova slide-a i brojačne vrijednosti slide-a u datoteku.

Vizualno to izgleda: <http://pastebin.com/HvdtXeKA>

Daljni korak: Povezati to sve s algoritmom koji će pratiti teme i polako završavati skriptu (napraviti jednostavih par predložaka koje korisnik može izabrati kao njegov tip prezentacije, a i mogućnost čistog unosa koordinata gdje se što nalazi)

Povezao sam većinu toga, sad još moram jasnije definirati predložke slide-ova i to sve isprobati na

kvalitetnijoj snimci predavanja, a onda napraviti **Probu 03** i dovršavati cijeli projekt.

Napravio početne probe, ali i dalje radim sa snimkom manje kvalitete a ne 1024×768, ali funkcionalnost algoritama je tu, još samo treba fino ukomponirati što bolju kvalitetu, jer nam output ovisi najviše o OCR-u, i nakon toga trebam još povezati sve preko modula i napraviti par poboljšanja (dodatne ideje) i onda ćemo definirati predloške slajdova te način kako ih unositi (i za to imam ideje).

Uspio sam posložiti sistem koji čita ono što treba, a ako ne uspije pokušava povećati sliku. Nešto kao double-check.

Uglavnom, rezultat je da mogu točno obraditi svaku snimku u formatu 1024×768. Neće biti potrebno povećavati na 1920×1080.

Nakon jednostavne optimizacije algoritma, vrijeme izvođenja je cca 1h i 30 min za snimku od 2-3 sata. – Zadnji rezultati

Planiram i dalje poboljšavati brzinu i učinkovitost algoritama te smanjivati pogreške OCR-a.

Jedan primjer output file-a za dio snimke 30-minuta, tj. 902 slike:

Legenda: BlueImage - Pojava plave slike, Something Else: Desktop, i sve ostalo, PowerPoint: Pojava powerpoint Sučelja, TitleSlide-a: Naslovi tema u samoj prezentaciji.

<http://pastebin.com/h1uYVzr>

7. tjedan (24.04.2016 - 01.05.2016)

Rad za vikend (koliko stignem):

- 1) Uređivanje, poboljšanja i implementacija user inputa.
- 2) Sređivanje GUI-a
- 3) Sređivanje dokumentacije, izrada mind-mape projekta

8. tjedan (01.05.2016 - 08.05.2016)

Nastavak:

- 1) Uređivanje, poboljšanja i implementacija user inputa.
- 2) Sređivanje GUI-a
- 3) Sređivanje dokumentacije, izrada mind-mape projekta

9. tjedan (09.05.2016 - 15.05.2016)

1) Unos početnih uvjeta

- Unos startne pozicije
- Unos plave slike
- Broj slika
- Avconv opcija
- Biranje Template-a

2) Napraviti 2 export modula

3) Bug Fixes, naming, documentation, how to tutorials

4) Izrada GUI-a

7) Proba 04 - Ubuntu 16, Tesseract

10. tjedan (15.05.2016 - 23.05.2016)

Ovaj tjedan nisam stigao radi obveza.

11. Tjedan (23.05.2016 - 29.05.2016)

1) Unos početnih uvjeta == **DONE**

- Unos startne pozicije
- Unos plave slike
- Broj slika
- Avconv opcija
- Biranje Template-a

2) Napraviti 2 export modula == **DONE**

3) Bug Fixes, naming, documentation, how to tutorials == **Partially Done**

4) Izrada GUI-a == **In process**

5) Proba 04 - Ubuntu 16, Tesseract == **In Process**

12. Tjedan (30.05.2016 - 05.06.2016)

- 1) Poboljšati algoritam za prepoznavanje slide-ova
- 2) Definirati opći template i napraviti par proba
- 3) Dovršiti dokumentaciju
- 4) Isprobati sve na Ubuntu 16

13. Tjedan (06.06.2016 - 12.06.2016)

Zbog nadolazećih ispita ne stižem baš previše raditi na skripti, ali ću sada dovršiti dokumentaciju i procesiranje u skripti. A onda ako ne stignem prije, nakon završnih ispita složiti kvalitetan GUI za sve ostalo.

U planu je i izvodi se:

- 1) Poboljšati algoritam za prepoznavanje slide-ova
- 2) Definirati opći template i napraviti par proba
- 3) Dovršiti dokumentaciju
- 4) Isprobati sve na Ubuntu 16

NOTES: 20-7-2016

Nakon par dana svladavanja Pythona i Tkinter Modula, ostalo mi je još odraditi neke sitnice.

Pa sam razmišljao za dodavanje baze podataka, u koju bi spremao podatke tijekom izvođenja programa kako se nebi izgubili ako se program sruši. (nestane struje ili nešto slično), pa će se jednostavnije moći i nastaviti određena obrada u bilo kojem trenutku.

Koliko će to biti pogodno za brzinu izvođenja isprobati ću vjerojatno sutra.
Sutra ću polako dovršavati cijelu skriptu i krenuti u izradu GUI-a kao osnovnog dijela aplikacije.

Isprobao sam i kako bi se napravio neki system independent .exe file, i našao sam način.
Još mi nakon završavanja svega preostaje samo igranje sa unosom template-a, izrada pokusa, probe na ubuntu a i općenito sa .exe aplikacijom i slično.

~~ NOTO C ~ ~

From:

<http://studentski-izvjestaji.zesoi.fer.hr/> - **Studentski izvještaji**

Permanent link:

http://studentski-izvjestaji.zesoi.fer.hr/doku.php?id=studenti:tibor_milic:tm_start



Last update: **2023/06/19 18:21**