

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



Εργασία Μαθήματος Ανάλυση Εικόνας

Τίτλος εργασίας	Απαλλακτική Εργασία του μαθήματος «Ανάλυση Εικόνας»
Όνομα φοιτητή - ΑΜ	Κωνσταντίνος-Ζώης Σύριος - Π16139 Νικόλαος Τόλης - Π16146
Ημερομηνία παράδοσης	30/03/2020

Περιεχόμενα

Γενική Περιγραφή	3
Περιβάλλον και Εργαλεία.....	3
Παρουσίαση Εργασίας	3
Γενικά	3
Ερώτημα i.	3
Ερώτημα ii.	4
Ερώτημα iii.	5
Ερώτημα iv.	5
Ερώτημα v.	6
Βιβλιογραφία	8

Γενική Περιγραφή

Το θέμα της εργασίας είναι ο αυτόματος χρωματισμός ασπρόμαυρης εικόνας με χρήση τεχνικών μηχανικής μάθησης. Για την υλοποίηση χρησιμοποιήσαμε την γλώσσα προγραμματισμού Python και βιβλιοθήκες αυτής (π.χ. cv2, sklearn).

Περιβάλλον και Εργαλεία

Χρησιμοποιήσαμε ως IDE το Community Edition του PyCharm (<https://www.jetbrains.com/pycharm>) έκδοση 2019.1.3. Ως γλώσσα επιλέξαμε την Python 3.7.4 και κάναμε χρήση των εξής βιβλιοθηκών:

- numpy
- opencv-python
- scikit-image (μετατροπή εικόνων, κατάτμηση εικόνων, φίλτρα)
- scikit-learn (KMeans, SVM)
- mahotas (SURF features)
- pickle (αποθήκευση και φόρτωση εκπαιδευμένων μοντέλων)
- urllib

Παρουσίαση Εργασίας

Γενικά

Ως dataset χρησιμοποιήσαμε ένα σύνολο εικόνων που απεικονίζουν μαργαρίτες, μικρού σχετικά μεγέθους ώστε η εκτέλεση του προγράμματος να γίνεται σχετικά γρήγορα.

Ερώτημα i.

Αναπαράσταση Εικόνας στον Χρωματικό Χώρο Lab

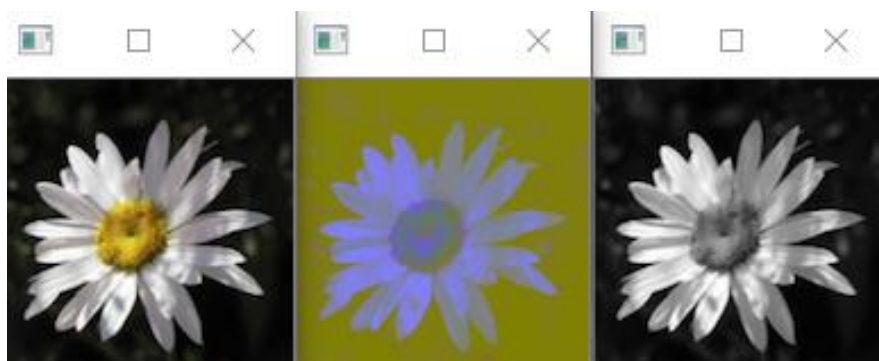
Για τη μετατροπή των εικόνων από και σε RGB, Grayscale και χρωματικό χώρο LAB χρησιμοποιούμε τις μεθόδους του αρχείου `image_functions.py`. Το προαναφερθέν αρχείο χρησιμοποιεί την βιβλιοθήκη cv2 για να “διαβάσει” εικόνες, να τις μετατρέψει και να τις εμφανίσει.

Ο χρωματικός χώρος LAB εκφράζει τα χρώματα ως τρεις τιμές:

- L για τη φωτεινότητα (μαύρο = 0, άσπρο = 100)
- A για πράσινο (αρνητικό) έως κόκκινο (θετικό)
- B από μπλε (αρνητικό) έως κίτρινο (θετικό)

Με $A = 0$, $B = 0$ αναπαρίσταται το πραγματικό ουδέτερο γκρι (*true neutral gray*)

Επίσης, στο αρχείο `image_functions.py` περιλαμβάνεται και η συνάρτηση που δημιουργεί και εμφανίζει το τελικό αποτέλεσμα της εργασίας.



1 Εμφάνιση Εικόνας σε RGB, LAB, Grayscale

Ερώτημα ii.

Διακριτοποίηση του Χρωματικού Χώρου Lab με βάση ένα σύνολο συναφών εικόνων εκπαίδευσης

Η διακριτοποίηση γίνεται από το αρχείο `colorspace_discretization.py` χρησιμοποιώντας KMeans clustering από την βιβλιοθήκη `sklearn`. Για να κάνουμε fit το KMeans μοντέλο, χρησιμοποιούμε ένα υποσύνολο του dataset (τις 13 από τις 14 εικόνες). Από τις εικόνες αυτές, αφού τις μετατρέψουμε στην LAB μορφή τους, εξάγουμε τις τιμές *a* και *b*. Με τις τιμές αυτές, γίνεται clustering σε 16 χρώματα.

Για εξοικονόμηση χρόνου γίνεται αποθήκευση του εκπαιδευμένου μοντέλου με τη χρήση της βιβλιοθήκης `pickle` και φόρτωση του όταν χρειάζεται.

```
TRAINING
Fitting kmeans...
Done fitting!
Saved kmeans!
```

2 Στιγμιότυπο εκτέλεσης του προγράμματος για το δεύτερο ερώτημα

Ερώτημα iii.

Κατάτμηση Εικόνας σε Superpixels σύμφωνα με τον αλγόριθμο SLIC

Η κατάτμηση της κάθε εικόνας σε Superpixels γίνεται με τις μεθόδους του αρχείου `slic_superpixels.py`. Χρησιμοποιούμε την υλοποίηση του αλγορίθμου SLIC από την βιβλιοθήκη `skimage.segmentation` και εμφανίζουμε το αποτέλεσμα με τη βοήθεια της μεθόδου `mark_boundaries` από την ίδια βιβλιοθήκη.

Ο αλγόριθμος Simple Linear Iterative Clustering (SLIC) ομαδοποιεί μεμονωμένα pixels βάσει χρωματικής ομοιότητας και απόστασης στον χρωματικό χώρο LAB.



3 Αποτέλεσμα Αλγορίθμου SLIC για 100 Superpixels

Ερώτημα iv.

Εξαγωγή Χαρακτηριστικών Υφής (SURF Features & Gabor Features) ανά Super Pixel

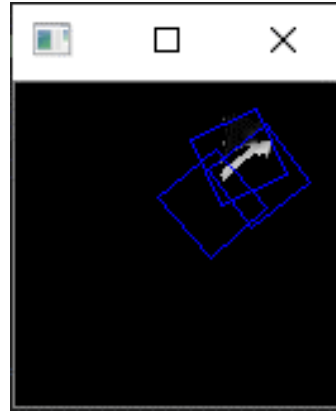
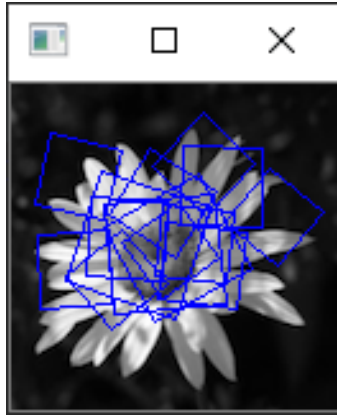
a) SURF Features

Η εξαγωγή των χαρακτηριστικών SURF γίνεται από τις μεθόδους του αρχείου `extract_features.py`. Χρησιμοποιούμε την υλοποίηση από την βιβλιοθήκη `mahotas`. Για την εξαγωγή των χαρακτηριστικών απαιτείται μετατροπή της εικόνας σε Grayscale.

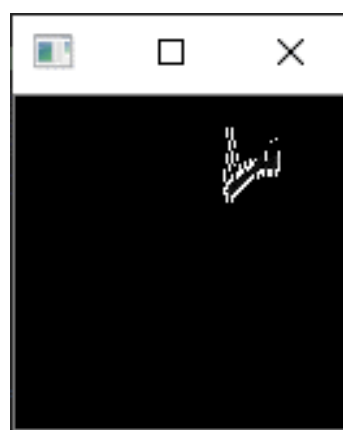
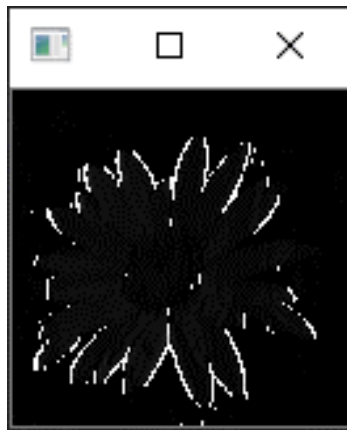
b) Gabor Features

Η εξαγωγή των χαρακτηριστικών Gabor γίνεται από τις μεθόδους του αρχείου `extract_features.py`. Χρησιμοποιούμε την υλοποίηση από την βιβλιοθήκη `skimage.filters`. Για την εξαγωγή των χαρακτηριστικών απαιτείται μετατροπή της εικόνας σε Grayscale.

Η παραπάνω διαδικασία μπορεί να γίνει τόσο σε ολόκληρες εικόνες όσο και σε μεμονωμένα Superpixels, διαχωρισμός των οποίων γίνεται από το αρχείο `slic_superpixels.py`.



4 SURF Χαρακτηριστικά σε ολόκληρη την εικόνα και σε μεμονωμένο Superpixel



5 Gabor Χαρακτηριστικά σε ολόκληρη την εικόνα και σε μεμονωμένο Superpixel

Ερώτημα ν.

Εκμάθηση Τοπικών Μοντέλων Πρόγνωσης Χρώματος με Χρήση Ταξινομητών SVM

Το τελευταίο αυτό υποερώτημα υλοποιείται στο αρχείο svm.py, στην μέθοδο fit_and_save_svm.

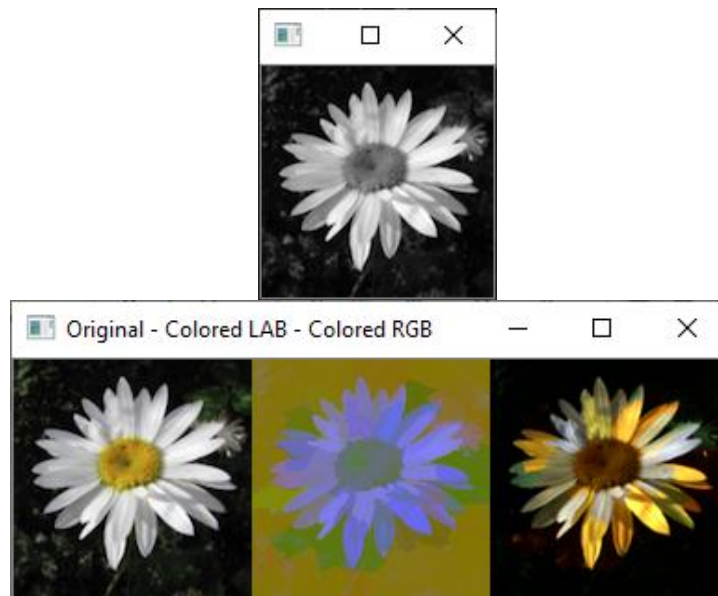
Το SVM που χρησιμοποιούμε είναι το SVC με την χρήση “balanced” βαρών για την κάθε κλάση. Για να κάνουμε fit το SVM, χρησιμοποιούμε το σύνολο των εικόνων που χρησιμοποιήσαμε και για τα KMeans. Ως features χρησιμοποιήσαμε τα χαρακτηριστικά SURF και Gabor του κάθε superpixel της κάθε εικόνας, που υπολογίζονται ως ο μέσος όρος των αντίστοιχων χαρακτηριστικών των επιμέρους pixels. Στην συνέχεια, υπολογίζουμε το κυρίαρχο χρώμα του superpixel (κάνοντας predict το κάθε pixel με το εκπαιδευμένο KMeans μοντέλο μας και χρησιμοποιούμε το χρώμα που επιστρέφεται περισσότερες φορές) την ετικέτα του οποίου (από το εκπαιδευμένο KMeans) την χρησιμοποιούμε ως ετικέτα των χαρακτηριστικών για το συγκεκριμένο superpixel.

Για εξοικονόμηση χρόνου γίνεται αποθήκευση του εκπαιδευμένου μοντέλου με τη χρήση της βιβλιοθήκης pickle και φόρτωση του όταν χρειάζεται.

```
Dataset downloaded already! If it is not delete 'Dataset' folder and re-run the program.  
TRAINING  
Fitted kmeans already saved -- If you wish to fit again stop execution, delete the file and rerun!  
Started SLIC segmentation...  
SLIC segmentation finished!  
Extracting SURF features...  
SURF extraction finished!  
Extracting Gabor features...  
Gabor extraction finished!  
Started fitting SVM...  
Finished fitting SVM!
```

6 Στιγμιότυπο από την εκτέλεση του προγράμματος μετά την εκπαίδευση του SVM

Για την εμφάνιση των αποτελεσμάτων εξάγονται τα χαρακτηριστικά SURF και Gabor από μια καινούργια Grayscale εικόνα τα οποία τα χρησιμοποιούμε για να κάνουμε predict από το εκπαιδευμένο SVM. Ύστερα από την μετατροπή των ετικετών των προβλέψεων στις αντίστοιχες a και b τιμές, γίνεται ένωση των L τιμών από την grayscale εικόνα με τις αντίστοιχες a, b τιμές.



7 Αποτελέσματα από την εκτέλεση του προγράμματος

Βιβλιογραφία

1. "A Framework for Using Custom Features to Colorize Grayscale Images"
<https://kb.osu.edu/handle/1811/76395>
2. "Machine Learning Methods for Automatic Image Colorization"
https://www.lri.fr/~gcharpia/colorization_chapter.pdf?fbclid=IwAR3QsUWIOBk9SRkP-Fespyy0Ekk8L7Uo2igtIsJfmK8INAJPF-UJCGWhfI
3. "Grayscale Image Colorization Using Machine Learning Techniques"
<https://cs.uwaterloo.ca/~zfrenett/CS886-Project.pdf>
4. "SURF: Speeded Up Robust Features"
<https://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
5. "Image Colorization Using Similar Images"
<https://people.cs.clemson.edu/~jzwang/ustc13/mm2012/p369-gupta.pdf>
6. "SLIC Superpixels"
http://www.kev-mith.com/papers/SLIC_Superpixels.pdf?fbclid=IwAR2XEHWICmCUSoHc1AucxUWZh-BsTiTQDawFm37lx4BwOYAK9-B3C-KbW1s