



Ανάλυση Εικόνας

Απαλλακτική εργασία εξαμήνου

Π16036 – Ιωαννίδης Παναγιώτης

Π16097 – Νίκας Διονύσιος

Π16112 – Παραβάντης Αθανάσιος

Περιεχόμενα

Εκφώνηση	3
Εισαγωγή.....	4
Προσδιορισμός χρωματικού χώρου	5
Διακριτοποίηση χρωματικού χώρου	6
Κατάτμηση σε superpixels	7
Χαρακτηριστικά SURF	8
Χαρακτηριστικά Gabor	9
Κατασκευή dataset	10
Εκπαίδευση μοντέλου SVM	11
Χρωματισμός ασπρόμαυρης εικόνας	11
Παρατηρήσεις.....	12

Εκφώνηση

Θέμα: Αυτόματος Χρωματισμός Ασπρόμαυρης Εικόνας με Χρήση Τεχνικών Μηχανικής Μάθησης.

Να υλοποιήσετε τις απαραίτητες αλγοριθμικές διαδικασίες για τον αυτόματο χρωματισμό μιας ασπρόμαυρης εικόνας. Συγκεκριμένα, θα πρέπει να αναπτύξετε κώδικα σε περιβάλλον Matlab ή Python για την διενέργεια των παρακάτω υπολογιστικών δραστηριοτήτων:

- i. Αναπαράσταση Εικόνας στον Χρωματικό Χώρο Lab
- ii. Διακριτοποίηση του Χρωματικού Χώρου Lab με βάση ένα σύνολο συναφών εικόνων εκπαίδευσης.
- iii. Κατάτμηση Εικόνας σε Superpixels σύμφωνα με τον αλγόριθμο SLIC.
- iv. Εξαγωγή Χαρακτηριστικών Υφής (SURF Features & Gabor Features) ανά Super Pixel.
- v. Εκμάθηση Τοπικών Μοντέλων Πρόγνωσης Χρώματος με Χρήση Ταξινομητών SVM
- vi. Εκτίμηση Χρωματικού Περιεχομένου Ασπρόμαυρης Εικόνας με Χρήση Αλγορίθμων Κοπής Γραφημάτων.

Πηγές:

- “A Framework for Using Custom Features to Colorize Grayscale Images”
<https://kb.osu.edu/handle/1811/76395>
- “Machine Learning Methods for Automatic Image Colorization”
http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/Colorization_main_6334%5b0%5d.pdf
- “Grayscale Image Colorization Using Machine Learning Techniques”
<https://cs.uwaterloo.ca/~zfrenett/CS886-Project.pdf>
- “SURF: Speeded Up Robust Features”
<https://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
- “Image Colorization Using Similar Images”
<https://people.cs.clemson.edu/~jzwang/ustc13/mm2012/p369-gupta.pdf>
- “SLIC Superpixels”
<https://people.cs.clemson.edu/~jzwang/ustc13/mm2012/p369-gupta.pdf>

Για την συγκεκριμένα εργασία μπορείτε να εργασθείτε σε ομάδες των τριών φοιτητών το πολύ. Παράλληλα με τον κώδικα της εργασίας θα πρέπει να παραδώσετε αναλυτική τεκμηρίωση της υλοποίησής σας καθώς και στιγμιότυπα ορθής εκτέλεσης του κώδικα. Η εργασία θα εξεταστεί προφορικά.

Εισαγωγή

Στην εργασία του μαθήματος «Ανάλυση Εικόνας» καλούμαστε να υλοποιήσουμε ένα πρόγραμμα, το οποίο θα δέχεται ως είσοδο μία (ή περισσότερες) εικόνες, μία ασπρόμαυρη εικόνα και ως έξοδο θα δίνεται μία προσεγγιστικά χρωματισμένη έκδοση της ασπρόμαυρης εικόνας. Για την επίλυση του προβλήματος, είναι απαραίτητη η ανάπτυξη ενός αλγορίθμου μηχανικής μάθησης, ο οποίος θα συλλέγει αξιοσημείωτα χαρακτηριστικά έγχρωμων εικόνων, και με βάση αυτών, θα εκτελεί τον χρωματισμό ασπρόμαυρων εικόνων. Οι τεχνολογικές απαιτήσεις μιας τέτοιας διαδικασίας είναι αρκετά μεγάλες, επομένως κρίνεται απαραίτητη η χρήση βιβλιοθηκών για στοιχειώδη επεξεργασία εικόνων αλλά και δημιουργία μοντέλων μηχανικής μάθησης.

Μετά από ανάλυση των απαιτήσεων της εργασίας, αποφασίσαμε να βασιστούμε στη γλώσσα προγραμματισμού Python και συγκεκριμένα την έκδοση 3.7. Επιπλέον, χρησιμοποιήσαμε τέσσερις σημαντικές βιβλιοθήκες, το OpenCV για την επεξεργασία των εικόνων, το numpy για την αναπαράσταση πινάκων, το skimage για εφαρμογή έτοιμων αλγορίθμων και το sklearn για την ανάπτυξη μοντέλων μηχανικής μάθησης. Συγκεκριμένα, μέσω του OpenCV καταφέρνουμε να αλλάζουμε τον χρωματικό χώρο εικόνων ευέλικτα, να βρίσκουμε χαρακτηριστικά SURF, Gabor και να αποθηκεύουμε αποτελέσματα επεξεργασίας σε εικόνες. Με το skimage επιτυγχάνουμε να εφαρμόσουμε τον αλγόριθμο SLIC σε χρωματισμένες και ασπρόμαυρες εικόνες αλλά και να απομονώσουμε superpixels. Τέλος, με το sklearn δημιουργούμε ένα μοντέλο SVM το οποίο χρησιμοποιείται στην εκμάθηση των χρωμάτων.

Η δομή των φακέλων και των αρχείων της εργασίας έχει ως εξής:

- Οι φωτογραφίες “source.jpg” και “target.jpg” χρησιμοποιούνται ως υπόδειγμα για τα ερωτήματα τις εργασίας.
- Το αρχείο “requirements.txt” περιέχει όλες τις βιβλιοθήκες που πρέπει να εγκατασταθούν στη Python για την εκτέλεση του προγράμματος.
- Το αρχείο “image_processor.py” περιέχει μια κλάση ImageProcessor που υλοποιεί όλη τη λογική, για τη κάλυψη των απαιτήσεων.
- Το αρχείο “app.py” είναι το σημείο έναρξης του προγράμματος και δημιουργεί ένα αντικείμενο ImageProcessor που θα εκτελέσει την επεξεργασία των εικόνων.
- Ο φάκελος “temp” και σχετικοί υποφάκελοι δημιουργούνται κατά τη διάρκεια εκτέλεσης του προγράμματος, ως προσωρινός αποθηκευτικός χώρος για εικόνες αρχικών, ενδιάμεσων και τελικών σταδίων επεξεργασίας.

Σημειώνεται πως για την ομαλή εκτέλεση και εγκατάσταση των βιβλιοθηκών στη Python, έχουμε δημιουργήσει ένα virtual environment βασισμένο στην έκδοση 3.7.3. Το OpenCV επειδή δεν περιέχει όλους τους αλγορίθμους που είχε σε παλαιότερες εκδόσεις, το έχουμε «χτίσει» ξεχωριστά.

Προσδιορισμός χρωματικού χώρου

Το πρώτο βήμα είναι η φόρτωση των εικόνων source και target. Η εικόνα source θεωρείται έγχρωμη και χρησιμοποιείται ως πρότυπο για τη διαδικασία χρωματισμού. Από τον χρωματικό χώρο RGB μεταβαίνουμε στον χρωματικό χώρο CIELAB. Η εικόνα target θεωρείται και αυτή έγχρωμη, όμως μετατρέπεται σε ασπρόμαυρη για λόγους προσομοίωσης.

Οι παρακάτω φωτογραφίες δείχνουν τα στάδια μετατροπής της εικόνας source από RGB σε LAB, αποθηκεύοντας ξεχωριστά τα κανάλια L, a και b.



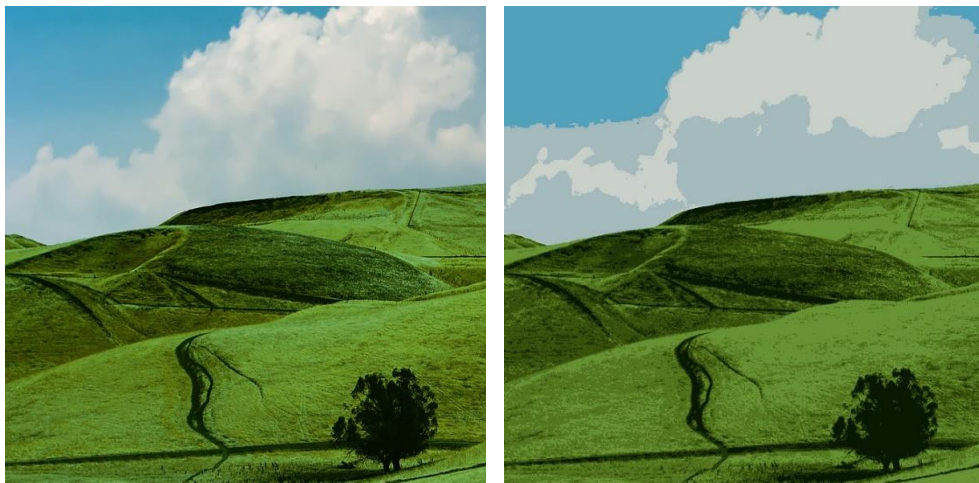
Η εικόνα target μετατρέπεται από RGB σε greyscale όπως φαίνεται παρακάτω.



Διακριτοποίηση χρωματικού χώρου

Το δεύτερο βήμα είναι η διακριτοποίηση του χρωματικού χώρου για την εικόνα source, ώστε να μπορέσουμε να σχηματίσουμε το dataset στα επόμενα βήματα. Η διαδικασία αυτή γίνεται με την εφαρμογή του αλγορίθμου K-Means για όλα τα LAB pixels της εικόνας. Ουσιαστικά, τα κεντροειδή αναπαριστούν ένα χρώμα και τα διάχυτα pixel αντιστοιχούνται με ένα από αυτά. Με αυτό το τρόπο, το αποτέλεσμα μας είναι μια εικόνα με διακριτό αριθμό χρωμάτων.

Παρακάτω απεικονίζονται τα αποτελέσματα του αλγορίθμου για την source εικόνα.



Αξίζει να σημειωθεί πως όσο περισσότερα κεντροειδή υπάρχουν στον αλγόριθμο, τόσο καλύτερη θα είναι και η ποιότητα της παραγόμενης φωτογραφίας. Στη δική μας περίπτωση χρησιμοποιούμε 8 κεντροειδή.

Κατάτμηση σε superpixels

Το τρίτο βήμα για το σχηματισμό του dataset είναι η κατάτμηση των source και target σε superpixels. Με την εφαρμογή του αλγορίθμου SLIC αυτό γίνεται εύκολα και γρήγορα. Ο στόχος αυτού του βήματος είναι να τεμαχίσουμε τις εικόνες σε περιοχές ενδιαφέροντος έτσι ώστε να είναι εύκολη η αντιστοίχιση των χρωμάτων. Για παράδειγμα, ένα δέντρο σε μια από τις φωτογραφίες μπορεί να είναι ένα superpixel γιατί αυτό υποδηλώνει μετέπειτα πως πρέπει να χρωματιστεί με κάποια απόχρωση του πράσινου.

Παρακάτω παρουσιάζονται τα αποτελέσματα εφαρμογής του αλγορίθμου στην source εικόνα.

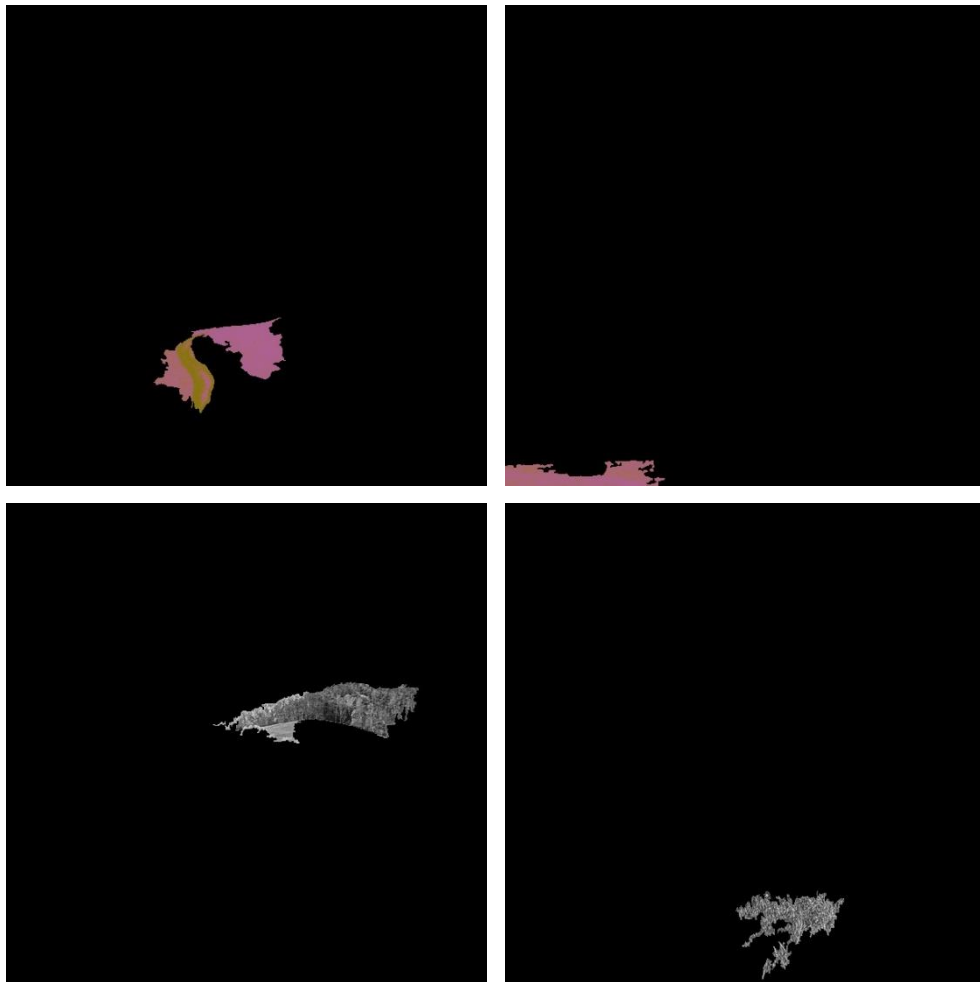


Για την εικόνα target εφαρμόζεται η αντίστοιχη διαδικασία στην greyscale έκδοση της.



Παρατηρούμε πως για την εικόνα source χρησιμοποιείται η κβαντισμένη μορφή της. Αυτό γίνεται διότι όταν αντιστοιχείται ένα χρώμα για κάθε superpixel, υπολογίζεται ο μέσος όρος των κβαντισμένων χρωμάτων μέσα στο superpixel. Η κατάτμηση γίνεται στην αρχική εικόνα αλλά αναπαρίσταται στην κβαντισμένη.

Αφού υπολογισθούν οι περιοχές των superpixel, αποθηκεύονται ξεχωριστά με τη βοήθεια μάσκας. Ενδεικτικά παραθέτουμε μερικά superpixel και από τις δυο εικόνες.



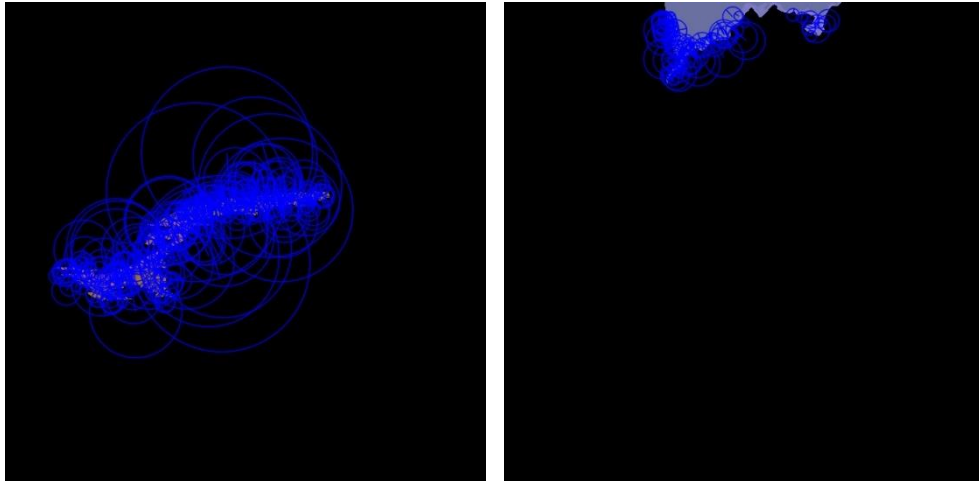
Παρατηρούμε πως τα superpixel της εικόνας source αποθηκεύονται με διαφορετικά χρώματα, επειδή έχουν μετατραπεί και κβαντιστεί στο χρωματικό χώρο LAB. Αυτό δεν ισχύει για την εικόνα target, επειδή παίρνουμε μόνο την greyscale έκδοσή της.

Για τα superpixel ορίζεται ως ανώτατο όριο 100 segments, το sigma 1 και το compactness 10 για την εικόνα source και 0.1 για την εικόνα target.

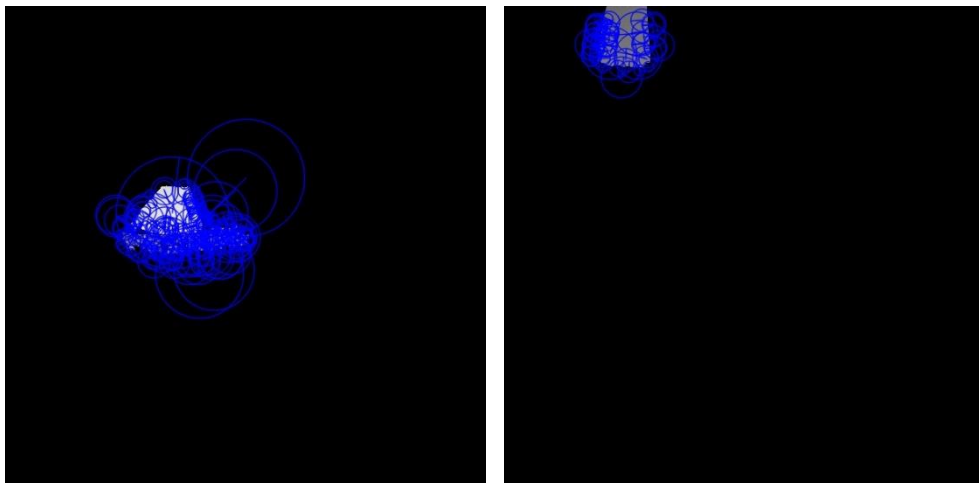
Χαρακτηριστικά SURF

Στο τέταρτο βήμα, τα χαρακτηριστικά SURF υπολογίζονται εύκολα με την βοήθεια της βιβλιοθήκης. Μας επιτρέπουν να προσδιορίσουμε αντικείμενα στα superpixel αλλά και τα «σύνορα» τους. Αυτές οι περιοχές ενδιαφέροντος μας βοηθούν στην κατασκευή των διανυσμάτων του συνόλου εκπαίδευσης.

Παρακάτω παραθέτονται για την εικόνα source δύο κβαντισμένα superpixel στο χρωματικό χώρο LAB μαζί με keypoints που προκύπτουν από τα χαρακτηριστικά SURF.



Για την εικόνα target έχουμε δύο superpixel σε greyscale μαζί με τα SURF χαρακτηριστικά.

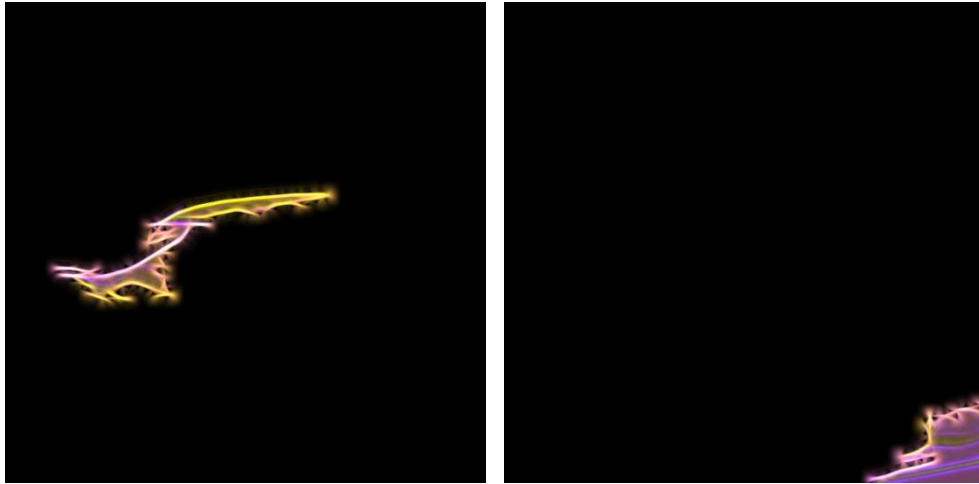


Χαρακτηριστικά Gabor

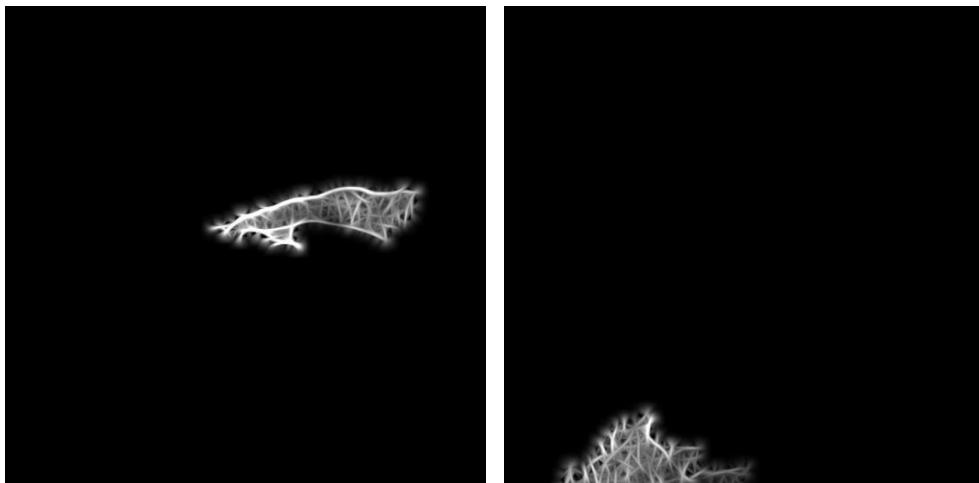
Στο πέμπτο βήμα, καλούμαστε να υπολογίσουμε τα χαρακτηριστικά Gabor δημιουργώντας παραδειγματικούς kernel και εφαρμόζοντάς τους στις φωτογραφίες για εξαγωγή χαρακτηριστικών υφής. Στην δική μας περίπτωση, δημιουργήσαμε 16 kernels αλλάζοντας κάθε φορά τη παράμετρο theta για διαφορετικές κατευθύνσεις.

Τα χαρακτηριστικά Gabor υπολογίζονται για κάθε superpixel ξεχωριστά, δημιουργώντας κάθε φορά 16 διαφορετικές εικόνες που είναι αποτέλεσμα εφαρμογής δισδιάστατου φίλτρου στη φωτογραφία μαζί με τον εκάστοτε kernel.

Παρακάτω απεικονίζονται για την source εικόνα, δύο superpixel σε LAB χρωματικό χώρο, αφού έχουν περάσει από φίλτρα των Gabor kernel.



Για την target εικόνα παρομοίως, έχουμε δύο superpixel σε greyscale αφού έχουν περάσει από τα ίδια φίλτρα των Gabor kernel.



Κατασκευή dataset

Για να δημιουργήσουμε και να εκπαιδεύσουμε ένα SVM πρέπει πρώτα να συλλέξουμε τα απαραίτητα δεδομένα που θα μας βοηθήσουν στη διαδικασία εκπαίδευσης. Τα προηγούμενα βήματα επεξεργασίας των εικόνων μας παρέχουν χρήσιμες πληροφορίες. Τόσο για την εικόνα source όσο και για την εικόνα target, το dataset εκπαίδευσης παραμένει το ίδιο. Αυτό που αλλάζει, είναι η απουσία των χρωμάτων label στην ασπρόμαυρη φωτογραφία.

Για την εικόνα source, θέλουμε να προσδιορίσουμε τα SURF και Gabor χαρακτηριστικά για κάθε superpixel καθώς και τον μέσο όρο των κβαντισμένων χρωμάτων μέσα σε αυτό. Ο σκοπός μας είναι να καταλήξουμε με ένα διάνυσμα χαρακτηριστικών της μορφής [64 τιμές SURF... 16 τιμές Gabor...] και μια τιμή label [index χρώματος]. Παρομοίως, για την εικόνα target, κατασκευάζονται διανύσματα για κάθε superpixel.

Οι τιμές label για τη χρωματισμένη εικόνα, υπολογίζονται με τη μέση τιμή των αποχρώσεων a και b διατρέχοντας όλα τα pixel των superpixel. Καταλήγουμε σε δύο νούμερα τα οποία πρέπει να αντιστοιχήσουμε με ένα κβαντισμένο χρώμα. Χρησιμοποιώντας το μοντέλο K-Means όπως και πριν καταφέρνουμε να πάρουμε το κεντροειδές που βρίσκεται πιο κοντά σε αυτές τις τιμές. Έτσι, προσδιορίζουμε ένα χαρακτηριστικό χρώμα ως label για το εκάστοτε superpixel.

Για τα χαρακτηριστικά SURF και Gabor, υπολογίζουμε τη μέση τιμή των «descriptor» και τη μέση τιμή της φιλτραρισμένης εικόνας αντίστοιχα. Τα αποτελέσματα ενώνονται όλα σε ένα διάνυσμα που αντιπροσωπεύει τα χαρακτηριστικά των superpixel. Πριν τελειώσει η διαδικασία, κανονικοποιούμε τις τιμές και αντιστοιχίζουμε τα διανύσματα με το αντίστοιχο χρώμα label. Έτσι, είμαστε έτοιμοι να ξεκινήσουμε την εκπαιδευτική διαδικασία και έπειτα να χρωματίσουμε την εικόνα.

Εκπαίδευση μοντέλου SVM

Αφού έχουμε έτοιμο το σύνολο εκπαίδευσης με τα διανύσματα και τα labels, δημιουργούμε ένα SVM για τη πρόβλεψη χρωμάτων και μετέπειτα τον χρωματισμό της εικόνας. Τα χαρακτηριστικά SURF, Gabor της source εικόνας χρησιμοποιούνται ως είσοδο και τα χρώματα labels χρησιμοποιούνται ως επιθυμητή έξοδο. Ανάλογα την εικόνα source και target, επιτυγχάνουμε αρκετά συχνά το accuracy να περνάει το 70-80%, επομένως τα αποτελέσματα μπορούν να θεωρηθούν ικανοποιητικά.

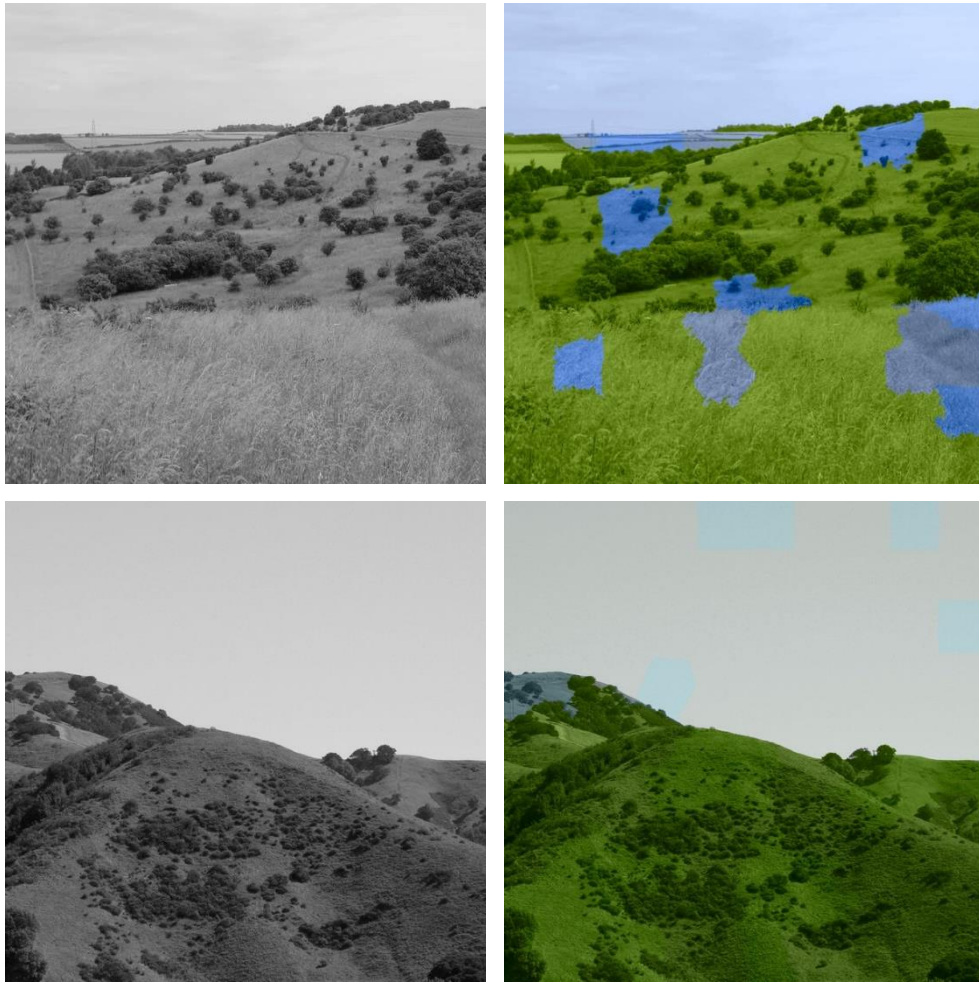
Χρωματισμός ασπρόμαυρης εικόνας

Χρησιμοποιώντας το SVM, τροφοδοτούμε ως είσοδο το σύνολο εκπαίδευσης της εικόνας target και αποθηκεύουμε τις τιμές χρωμάτων που προβλέφθηκαν.

Παρακάτω εμφανίζονται τα αποτελέσματα χρωματισμού μιας ενδεικτικής εικόνας:



Άλλες εικόνες που χρησιμοποιηθήκαν:



Παρατηρούμε πως το SVM χρωματίζει τις εικόνες με μερικές ατέλειες αλλά ικανοποιητικά. Αυτό πολλές φορές οφείλεται στην αδυναμία του αλγορίθμου SLIC να τεμαχίσει σωστά την εικόνα αλλά και στην απουσία ενός μεγάλου dataset εικόνων.

Παρατηρήσεις

Για την εγκατάσταση της βιβλιοθήκης OpenCV ακολουθήσαμε τις οδηγίες του άρθρου:

<https://www.pyimagesearch.com/2018/08/15/how-to-install-opencv-4-on-ubuntu/>

Ο λόγος για τον οποίο χτίσαμε εμείς τη βιβλιοθήκη, είναι επειδή στις τελευταίες εκδόσεις έχουν αφαιρεθεί κάποιοι patented algorithms.