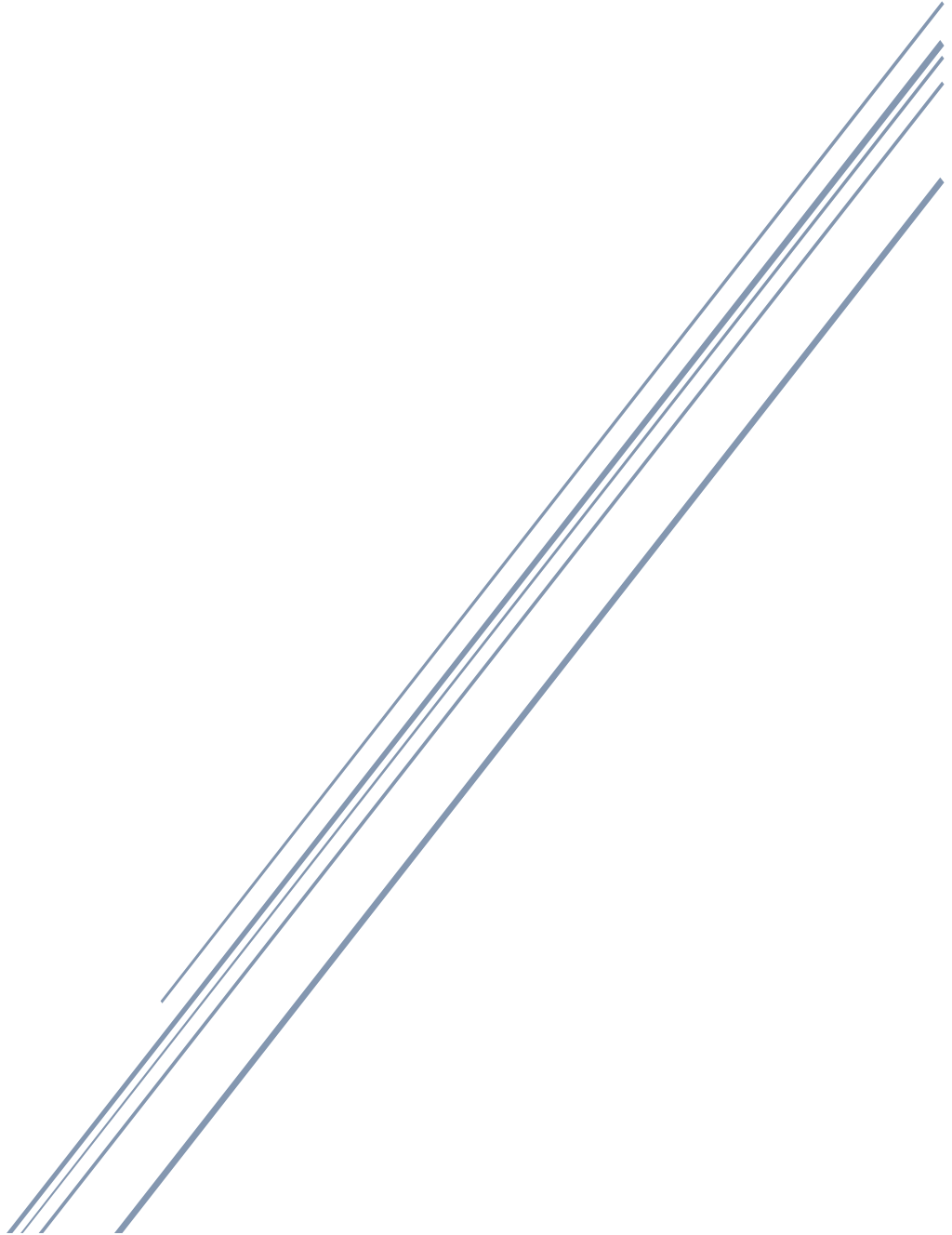


Ανάλυση Εικόνας

Απαλλακτική Εργασία

Ακαδημαϊκό Έτος 2020 - 2021



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Δημήτρης Ματσαγγάνης, Π17068
Αλέξανδρος Σκαρπέλος, Π17122

Περιεχόμενα

Εισαγωγή Εργασίας.....	3
Κεντρική Ιδέα Υλοποίησης.....	4
Απαραίτητα Εργαλεία για την Εκτέλεση.....	5
Δεδομένα Εκπαίδευσής Μοντέλου	6
Ερώτημα 1: Αναπαράσταση Εικόνας στον Χρωματικό Χώρο LAB	7
Ερώτημα 2: Διακριτοποίηση του Χρωματικού Χώρου LAB με βάση ένα σύνολο συναφών εικόνων εκπαίδευσης.....	9
Ερώτημα 3: Κατάτμηση Εικόνας σε Superpixels σύμφωνα με τον αλγόριθμο SLIC.....	10
Ερώτημα 4: Εξαγωγή Χαρακτηριστικών Υφής (SURF Features & Gabor Features) ανά Superpixel	12
SURF Features	12
Gabor Features	14
Ερώτημα 5: Εκμάθηση Τοπικών Μοντέλων Πρόγνωσης Χρώματος με Χρήση Ταξινομητών SVM.....	16
Κατασκευή Dataset.....	16
Εκπαίδευση SVM Μοντέλου	18
Χρωματισμός της Εικόνας Στόχου.	18
Αποτελέσματα Εκτέλεσης Υλοποίησης.....	20
Οδηγίες Εκτέλεσης Προγράμματος	21
Βιβλιογραφία.....	22
Περιεχόμενα Απεσταλμένου Αρχείου	23

Εισαγωγή Εργασίας

Απαλλακτική Εργασία 2020

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.

Κεντρική Ιδέα Υλοποίησης

Στην παρούσα ενότητα της εργασίας του μαθήματος «Ανάλυση Εικόνας», θα αναφερθούμε στην κεντρική ιδέα υλοποίησης, της εφαρμογής μας.

Αρχικά, από την εκφώνηση της εργασίας καλούμαστε να υλοποιήσουμε ένα πρόγραμμα, το οποίο θα δέχεται ως είσοδο μία (ή περισσότερες) εικόνες – που θα χρησιμοποιούνται ως βάση δεδομένων και παραδειγμάτων και μία ασπρόμαυρη εικόνα, ενώ ως έξοδο θα δίνεται μία προσεγγιστικά χρωματισμένη έκδοση της ασπρόμαυρης εικόνας.

Καίριο ρόλο στη προσέγγιση του χρωματισμού της ασπρόμαυρης εικόνας εισόδου έχουν οι έγχρωμες “εικόνες παραδείγματα” που έχουμε εισάγει ήδη στην εφαρμογή.

Ως εκ τούτου, για την επίλυση του προβλήματος, είναι απαραίτητη η ανάπτυξη ενός αλγορίθμου μηχανικής μάθησης.

Ο προαναφερθέν αλγόριθμος, θα συλλέγει τα αξιοσημείωτα χαρακτηριστικά έγχρωμων “εικόνων παραδειγμάτων”, και με βάση αυτών, θα εκτελεί τον χρωματισμό ασπρόμαυρων εικόνων εξόδου.

Δεδομένου αυτού, οι απαιτήσεις μιας τέτοιας διαδικασίας είναι αρκετά μεγάλες, επομένως κρίνεται απαραίτητη η χρήση βιβλιοθηκών της Python σχετικά με την επεξεργασία εικόνων αλλά και δημιουργία μοντέλων μηχανικής μάθησης.

Μέσω των κατάλληλων βιβλιοθηκών (θα αναλυθούν εκτενώς παρακάτω) σκοπός μας είναι να καταφέρνουμε να αλλάζουμε τον χρωματικό χώρο εικόνων ευέλικτα και να βρίσκουμε χαρακτηριστικά SURF, Gabor και έπειτα να αποθηκεύουμε αποτελέσματα επεξεργασίας σε νέες εικόνες.

Τέλος, με τον αλγόριθμο SLIC σε χρωματισμένες και ασπρόμαυρες εικόνες επιτυγχάνουμε να απομονώσουμε τα Superpixels και δημιουργούμε ένα μοντέλο SVM το οποίο χρησιμοποιείται στην εκμάθηση των χρωμάτων από το μοντέλο της εφαρμογής μας.

Απαραίτητα Εργαλεία για την Εκτέλεση

Για την επίλυση της εργασίας του μαθήματος και για την επιτυχή δημιουργία της εφαρμογής, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python (συγκεκριμένα η **έκδοση Python 3.7.9**).

Επιπρόσθετα, οι παρακάτω βιβλιοθήκες είναι απαραίτητες για την εκτέλεση του προγράμματος με βάση τις σχεδιαστικές προδιαγραφές του:

- ***numpy*** : για το χειρισμό των υπολογιστικών δεδομένων (αριθμητικών) από τη Python. Η συγκεκριμένη βιβλιοθήκη χρησιμοποιείτε στην υλοποίησή μας κυρίως για το χειρισμό και την αναπαράσταση πινάκων.
- ***cv2 (opencv-python)*** [έκδοση: 3.4.2.16] : για την εισαγωγή και επεξεργασία των εικόνων και γενικότερα για το κομμάτι της υπολογιστικής όρασης (computer vision).
- ***skimage*** : για εφαρμογή έτοιμων αλγορίθμων σχετικούς με την ανάλυση και την επεξεργασία εικόνας.
- ***sklearn*** : για την ανάπτυξη μοντέλων και εφαρμογών μηχανικής μάθησης.

Πιο συγκεκριμένα, μέσω της βιβλιοθήκης *cv2 (opencv-python)* καταφέρνουμε να αλλάζουμε τον χρωματικό χώρο εικόνων με ευελιξία, να βρίσκουμε χαρακτηριστικά SURF, Gabor, που είναι κρίσιμα για την επίλυση της άσκησης και να αποθηκεύουμε αποτελέσματα επεξεργασίας σε εικόνες.

Ακόμη με τη βιβλιοθήκη *skimage* επιτυγχάνουμε να εφαρμόσουμε κατάλληλα τον αλγόριθμο SLIC σε χρωματισμένες και ασπρόμαυρες εικόνες αλλά και να απομονώσουμε Superpixels.

Τέλος, με τη βιβλιοθήκη *sklearn* δημιουργούμε ένα μοντέλο SVM το οποίο χρησιμοποιείται στην εκμάθηση των χρωμάτων από το μοντέλο μας και με βάση τις εικόνες παραδείγματα που έχουν ήδη εισαχθεί.

Δεδομένα Εκπαίδευσής Μοντέλου

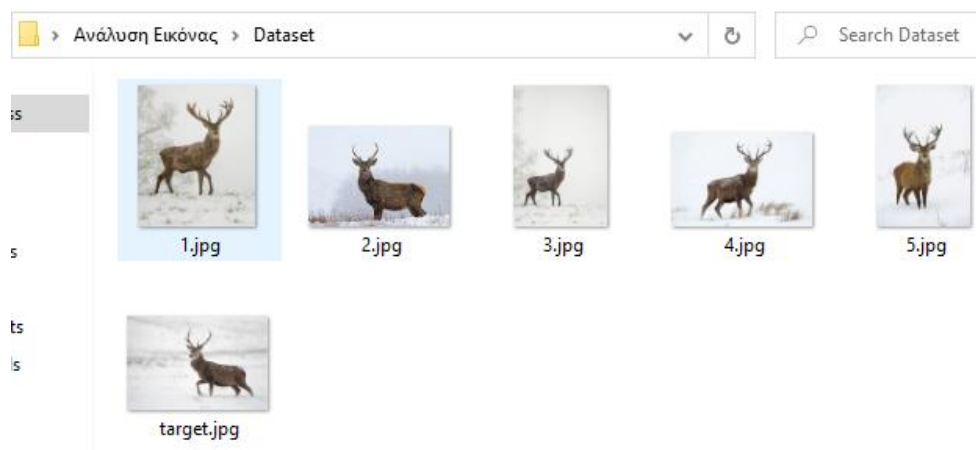
Όσο αναφορά τα δεδομένα, από τα οποία ο αλγόριθμος, θα συλλέγει τα χαρακτηριστικά των έγχρωμων “εικόνων παραδειγμάτων”, και με βάση αυτών, θα εκτελεί τη διαδικασία του χρωματισμού των ασπρόμαυρων εικόνων εξόδου, χρησιμοποιήσαμε ένα σύνολο εικόνων που απεικονίζουν ένα ελάφι σε ένα χιονισμένο τοπίο.



Εικόνα Παραδείγματος 1

Οι εικόνες θα είναι σχετικά μικρού μεγέθους ώστε η εκτέλεση του προγράμματος να γίνεται γρήγορα και όλες θα βρίσκονται στο φάκελο Dataset, εντός του φακέλου της εργασίας.

Ο υποφάκελος θα έχει την παρακάτω μορφή:



Εικόνα Αποτελεσμάτων 1

Ερώτημα 1: Αναπαράσταση Εικόνας στον Χρωματικό Χώρο LAB

Στη συγκεκριμένη ενότητα θα αναφερθούμε στην διαδικασία με την οποία θα γίνεται η αναπαράσταση της εικόνας στόχος “**target image**”, στο χρωματικό χώρο LAB.

Η αναπαράσταση σε αυτόν τον χρωματικό χώρο μας βοηθάει να απλοποιήσουμε το πρόβλημα μας διότι στον χρωματικό χώρο RGB θα είχαμε τρεις αγνώστους ενώ σε αυτό (LAB) έχουμε δυο.

Οι νέες – χρωματισμένες εικόνες αποθηκεύονται σε μια λίστα **lab_images**, καθώς θα τις χρησιμοποιήσουμε και σε επόμενα ερωτήματα. Αντίστοιχα, η εικόνα στόχος “**target**”, η οποία είναι αρχικά έγχρωμη, μετατρέπεται σε ασπρόμαυρη για λόγους ορθολογικής προσομοίωσης της ζητούμενης διαδικασίας.

Ειδικότερα, για τη μετατροπή των εικόνων από το χρωματικό χώρο RGB, στο χρωματικό χώρο LAB χρησιμοποιούμε την ακόλουθη εντολή:

```
cv2.cvtColor(rgb_image, cv2.BGR2LAB) (1)
```

Ενώ, η μετατροπή της εικόνας στόχου “target”, υλοποιείται με την παρακάτω εντολή:

```
cv2.cvtColor(target_image, cv2.BGR2GRAY) (2)
```

* Στις εντολές (1) και (2), οι παρακάτω μεταβλητές σημαίνουν:

- Η μεταβλητή **rgb_image** είναι η κάθε εικόνα εκπαίδευσης.
- Η μεταβλητή **target_image** είναι η εικόνα στόχος.

Ως εκ τούτου, για τις συναρτήσεις του συγκεκριμένου ερωτήματος και την παραπάνω διαδικασία, χρησιμοποιούμε την βιβλιοθήκη, εικονικής πραγματικότητας, cv2 για τη φόρτωση των εικόνων πηγής - source και στόχου - target, να τη μετατροπή και να την εμφάνισή τους.

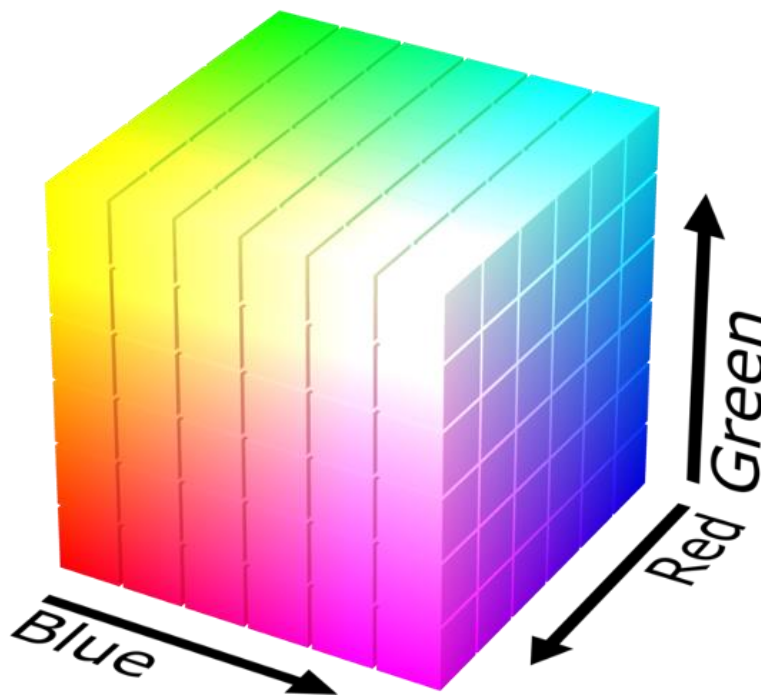
Ο χρωματικός χώρος LAB εκφράζει τα χρώματα ως τρεις τιμές:

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

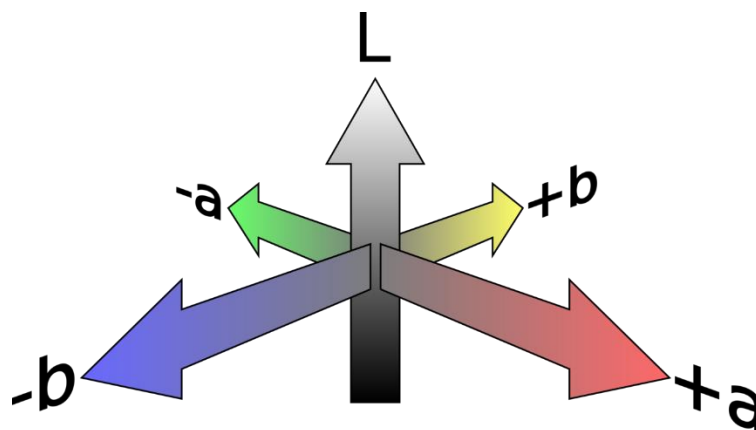
Με $A = 0$, $B = 0$ αναπαρίσταται το πραγματικό ουδέτερο γκρι (*true neutral gray*).

Τέλος, αξίζει να επισημανθεί το γεγονός ότι τα A , B είναι ανεξάρτητα από το L και την τιμή αυτού, ενώ όπως προαναφέρθηκε με την εν λόγω μετατροπή περιορίζουμε το πλήθος των αγνώστων για τους οποίους θα πρέπει ο αλγόριθμος να παρέχει μια εκτίμηση από τρεις σε δύο.

Ακολουθούν δύο εικόνες – αναφορές για τους χρωματικούς χώρους RGB και LAB:



Εικόνα Αποτελεσμάτων 2



Εικόνα Αποτελεσμάτων 3

Ερώτημα 2: Διακριτοποίηση του Χρωματικού Χώρου LAB με βάση ένα σύνολο συναφών εικόνων εκπαίδευσης

Στη παρούσα ενότητα θα γίνει εκτενής αναφορά στα ζητούμενα του 2^{ου} ερωτήματος της εργασίας.

Πιο συγκεκριμένα, η διακριτοποίηση του χρωματικού χώρου LAB με βάση ένα σύνολο εικόνων, ώστε να διαμορφωθεί το dataset, που θα χρησιμοποιηθεί στα επόμενα βήματα για την εκπαίδευση της παρούσας υλοποίησης, είναι το επόμενο ζητούμενο της εργασίας.

Η προαναφερθείσα διαδικασία, γίνεται με την εφαρμογή του αλγορίθμου K-Means Clustering (της βιβλιοθήκης sklearn) για όλα τα LAB pixels της εικόνας.

Για να κάνουμε ορθά fit το K-Means μοντέλο, χρησιμοποιούμε το σύνολο των χρωματισμένων εικόνων lab_images, αφού πρώτα τις έχουμε επεξεργαστεί μέσω του **reshape**.

Από το σύνολο των εικόνων, εξάγουμε τις τιμές A και B, ώστε να γίνει Clustering σε K χρώματα (η αλλαγή της τιμής του K θα γίνεται με μεγάλη ευκολία μέσω του κώδικα).

Αναλυτικότερα, τα κεντροειδή αναπαριστούν ένα χρώμα και τα διάχυτα pixel αντιστοιχούνται με ένα από αυτά (από τα διαθέσιμα). Με αυτό το τρόπο, το αποτέλεσμα μας είναι μια εικόνα με διακριτό αριθμό χρωμάτων σε pixels.

Πιο συγκεκριμένα, με αυτή τη διαδικασία της κβάντισης του χρωματικού χώρου θέλουμε να περιορίσουμε το σύνολο των πιθανών ετικετών (πεδίο ορισμού) στο οποίο θα προβλέπει και θα εκτελείται ο αλγόριθμος.

Τέλος, κρίνεται άξιο αναφοράς πως όσο μικρότερο είναι το K, τόσο πιο ακριβής είναι η ταξινόμηση (εύρεση των αποχρώσεων με ακρίβεια), αλλά τόσο χειρότερο θα είναι το αποτέλεσμα του χρωματισμού.

Στη συγκεκριμένη υλοποίηση το K έχει αρχικά οριστεί ίσο με 4.

Για την επίλυση του 2^{ου} ερωτήματος της άσκησης υλοποιήθηκε η συνάρτηση **image_quantization**.

Ερώτημα 3: Κατάτμηση Εικόνας σε Superpixels σύμφωνα με τον αλγόριθμο SLIC

Σε αυτή την ενότητα θα αναφερθούμε στην διαδικασία κατάτμησης της κάθε εικόνας (τόσο εικόνες εκπαίδευσης, όσο και στόχου) σε Superpixels, η οποία γίνεται με τη βοήθεια της συνάρτησης **slic_superpixels**

Πρόκειται για το τρίτο – κατά σειρά - βήμα για το σχηματισμό του dataset, που είναι η κατάτμηση των source και target σε Superpixels.

Ειδικότερα, για την παραπάνω υλοποίηση χρησιμοποιούμε τον αλγόριθμο SLIC από την βιβλιοθήκη *skimage* και συγκεκριμένα τη μέθοδο **segmentation**.

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

Με την εφαρμογή του αλγορίθμου SLIC η διαδικασία αυτή 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.

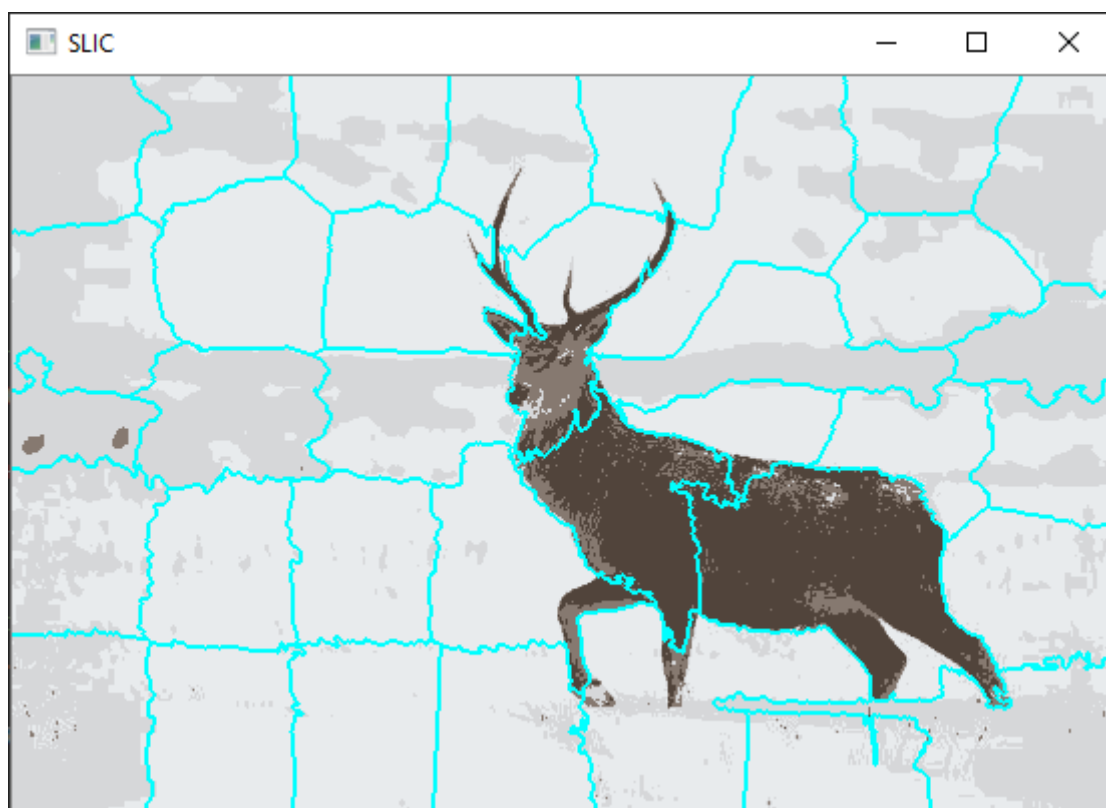
Για την εικόνα στόχου - target εφαρμόζεται η αντίστοιχη διαδικασία στην Greyscale έκδοση της.

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.

Εφόσον υπολογισθούν ορθολογικά οι περιοχές των superpixel, αποθηκεύονται ξεχωριστά με τη βοήθεια μάσκας. Η τελευταία (δλδ. η χρήση της μάσκας) βοηθάει στο να έχουμε τη δυνατότητα πρόσβασης σε κάθε ξεχωριστή κατάτμηση .

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 στόχου - target.

Ακολουθεί η εικόνα από την κατάτμηση SLIC:



Εικόνα Αποτελεσμάτων 4

```
Process: Extract SLIC Superpixels for the target image.  
Completed: Extract SLIC Superpixels for the target image.
```

Εικόνα Αποτελεσμάτων 5

Ερώτημα 4: Εξαγωγή Χαρακτηριστικών Υφής (SURF Features & Gabor Features) ανά Superpixel

Το επόμενο και τέταρτο βήμα της υλοποίησης μας αφορά την εξαγωγή ενός διάνυσματος χαρακτηριστικών για κάθε superpixel. Το διάνυσμα αυτό θα περιέχει **Surf Features** και **Gabor Features**.

Η διαδικασία εξαγωγής των **Surf Features** και **Gabor Features** θα αναλυθούν κατά βάθος στις δύο επερχόμενες ενότητες.

SURF Features

Αρχικά, τα χαρακτηριστικά υφής υπολογίζονται εύκολα με τη βοήθεια της συνάρτησης **Surf**. Ειδικότερα, με την εν λόγω διαδικασία γίνεται προσπάθεια για την εύρεση κύριων – αξιοσημείωτων χαρακτηριστικών (keypoints) εντός του κάθε superpixel.

Η εξαγωγή των συγκεκριμένων χαρακτηριστικών γίνεται με τη βοήθεια της συνάρτησης **surf_features**.

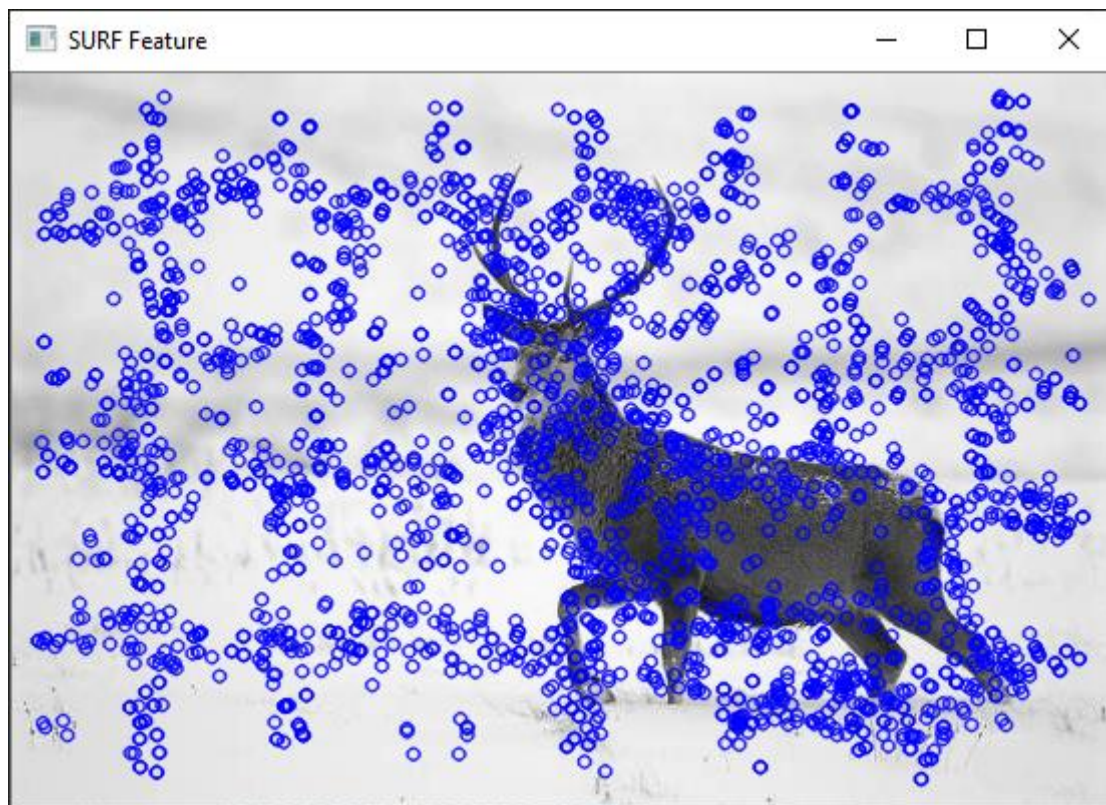
Όσο αναφορά τις εικόνες που ανήκουν στο σύνολο των εικόνων εκπαίδευσης, πρέπει να μετατραπούν σε πρώτη φάση σε grayscale.

Αξίζει να σημειωθεί το γεγονός πως, η διαδικασία για την εξαγωγή των συγκεκριμένων χαρακτηριστικών είναι ίδια και για τις εικόνες-παραδείγματα – εκπαίδευσης και για την εικόνα στόχο - target.

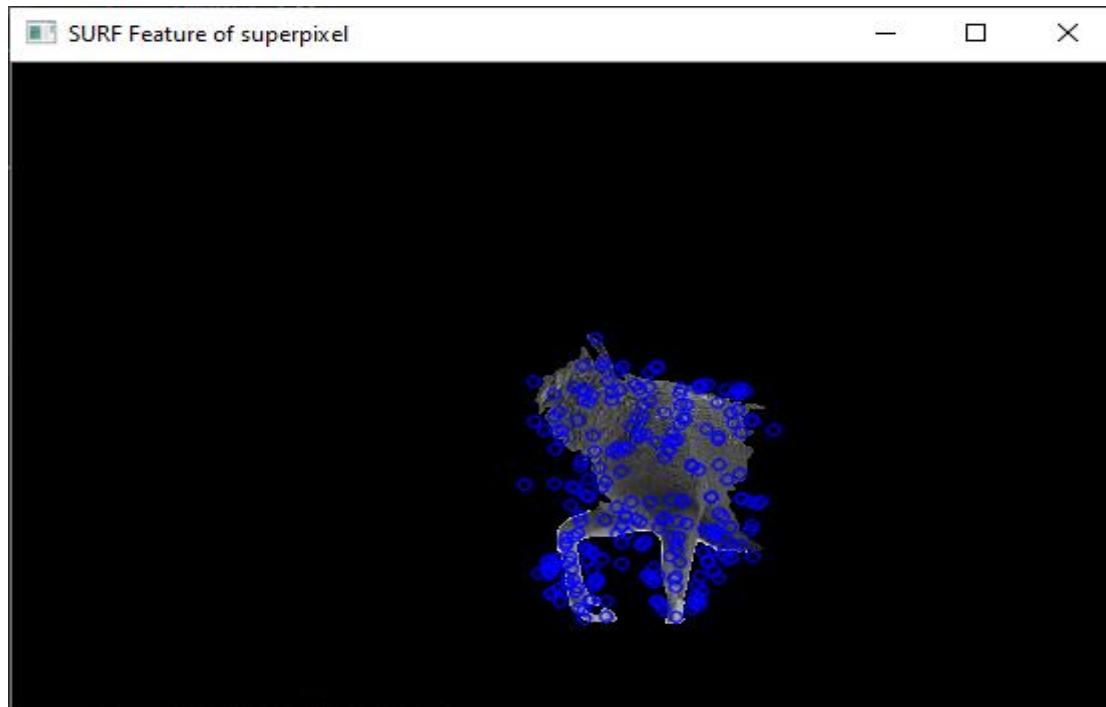
Με την μέθοδο της συνάρτησης **Surf, detect_and_compute**, η οποία δέχεται σαν όρισμα κάθε gray superpixel, όπως αυτό έχει προκύψει από την κατάτμηση της εικόνας (όπως αυτή έγινε στο ερώτημα 3), εντοπίζονται τα αξιοσημείωτα χαρακτηριστικά (keypoints), ενώ υπολογίζονται και οι descriptors.

Τέλος, τόσο τα keypoints, όσο και οι descriptors αποθηκεύονται σε διαφορετικούς πίνακες (arrays).

Παρακάτω υπάρχουν εικόνες για τα χαρακτηριστικά SURF:



Εικόνα Αποτελεσμάτων 6



Εικόνα Αποτελεσμάτων 7

```
Process: Extract SURF feature for the target image.  
Completed: Extract SURF feature for the target image.
```

Εικόνα Αποτελεσμάτων 8

Gabor Features

Ο υπολογισμός των χαρακτηριστικά **Gabor** (Gabor Features), γίνεται δημιουργώντας kernels και εφαρμόζοντάς τους (τους kernels) στο εκάστοτε superpixel για την εξαγωγή χαρακτηριστικών υφής.

Η δημιουργία των kernels γίνεται με τη συνάρτηση **build_kernels**. Αναλυτικότερα, για τη παρούσα διαδικασία δημιουργούνται **40** kernels από το συνδυασμό φίλτρων 8 rotations (theta) και 5 scales (lambda).

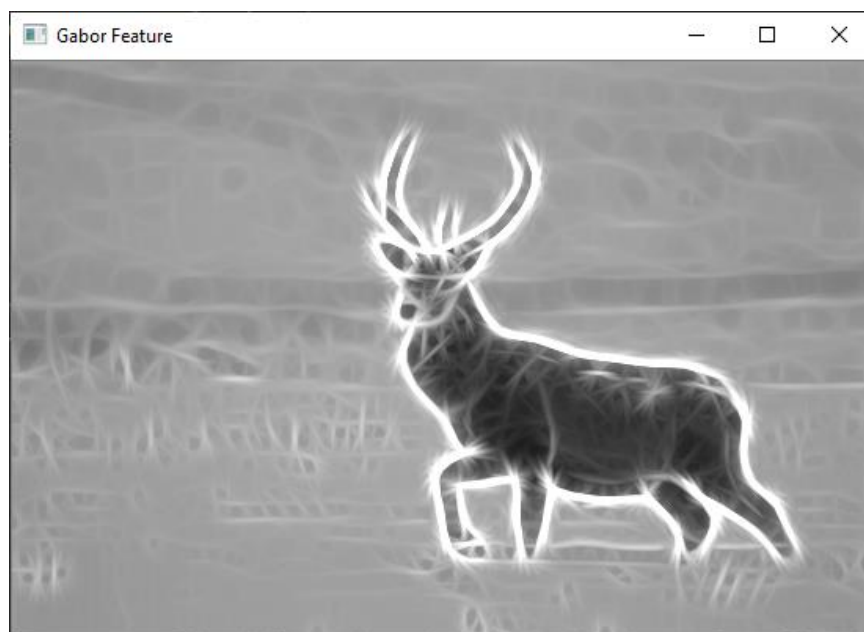
Στη συνέχεια, με τη συνάρτηση **gabor_features**, η οποία δέχεται σαν όρισμα τα superpixel μιας εικόνας, γίνεται η εξαγωγή των χαρακτηριστικών **Gabor**.

Αναφορικά με τις εικόνες εκπαίδευσης της υλοποίησης μας, πρέπει να μετατραπούν αρχικά σε grayscale.

Μέσω της συνάρτησης **apply_kernels** εφαρμόζονται τα kernels στο κάθε superpixel της εικόνας και εξάγονται τα χαρακτηριστικά υφής, τα οποία αποθηκεύονται σε ένα πίνακα (array).

Ακόμη, με τα **Gabor Features** το κάθε superpixel περνάει από μια ακολουθία συνελικτικών φίλτρων, η οποία εξομαλύνει τις τιμές φωτεινότητας (την τιμή του L από το χρηματικό χώρο LAB) σε διάφορες κλίμακες. Τα συνελικτικά φίλτρα, προσδιορίζονται από μια συνάρτηση, η οποία έχει ημιτονοειδή μορφή και παραμετροποιείται από τη φάση του ημιτόνου.

Ακολουθούν εικόνες για τα χαρακτηριστικά Gabor:



Εικόνα Αποτελεσμάτων 9



Εικόνα Αποτελεσμάτων 10



Εικόνα Αποτελεσμάτων 11

```
Process: Extract Gabor feature for target image.  
Completed: Extract Gabor feature for target image.
```

Εικόνα Αποτελεσμάτων 12

Ερώτημα 5: Εκμάθηση Τοπικών Μοντέλων Πρόγνωσης Χρώματος με Χρήση Ταξινομητών SVM

Στη συγκεκριμένη ενότητα θα αναφερθούμε στο τελευταίο ζήτημα της εργασίας, που είναι η **κατασκευή του Dataset**, η **εκπαίδευση ενός SVM μοντέλου** και τέλος ο **χρωματισμός της εικόνας στόχου**.

Συνεπώς το πέμπτο και τελευταίο ερώτημα αποτελείται από τρεις διαφορετικές, αλλά εξίσου σημαντικές διαδικασίες.

Σκοπός μας είναι η επεξήγηση αυτών των διαδικασιών στις ενότητες που θα ακολουθήσουν.

Κατασκευή Dataset

Ως πρωταρχικό βήμα για τη δημιουργία και την εκπαίδευση ενός SVM μοντέλου πρέπει πρώτα συλλέξουμε τα απαραίτητα δεδομένα που θα μας βοηθήσουν στη διαδικασία της εκπαίδευσης.

Τα παραπάνω βήματα επεξεργασίας των εικόνων μας παρέχουν χρήσιμα εργαλεία για την παρούσα κατάσταση. Τα εργαλεία αυτά είναι χρήσιμα τόσο για τις εικόνες εκπαίδευσης, όσο και για την εικόνα στόχου - target, το dataset εκπαίδευσης παραμένει το αρχικό.

Μια ουσιώδης διαφορά είναι η απουσία των χρωμάτων label στην ασπρόμαυρη φωτογραφία (Grayscale), τα οποία και θα προβλεφθούν από το SVM μοντέλο. Για τις εικόνες που ανήκουν στο σύνολο πηγής - source, θέλουμε να προσδιορίσουμε τα SURF και Gabor χαρακτηριστικά για κάθε superpixel, καθώς και τον μέσο όρο των κβαντισμένων χρωμάτων μέσα σε αυτό.

Σκοπός μας είναι να καταλήξουμε με ένα διάνυσμα χαρακτηριστικών της μορφής **[64 τιμές SURF... 40 τιμές Gabor]** και μια τιμή ετικέτας - label **[Index του χρώματος]**.

Κατά αναλογία, για την εικόνα target - στόχου κατασκευάζονται διανύσματα για κάθε superpixel αυτής.

Έπειτα, για τη διαδικασία κατασκευής του dataset για τις εικόνες παραδείγματα κρατάμε, σε πρώτη φάση, τα χρώματα που προέκυψαν από τη διαδικασία της κβάντισης, αποθηκεύοντας μόνο τις **A** και **B** τιμές, τιτλοδοτώντας αυτές με ένα id μέσα σε έναν πίνακα τιμών.

Το επόμενο στάδιο αυτής της υλοποίησης, είναι η εύρεση των κεντροειδών χρωμάτων για το εκάστοτε superpixel. Ειδικότερα, για καθένα superpixel βρίσκουμε τον μέσο όρο των LAB τιμών του και κάνει predict τις τιμές για βρει σε ποιο από τα κβαντισμένα χρώματα είναι πιο κοντά. Έπειτα, αποθηκεύονται οι A, B τιμές του κβαντισμένου χρώματος.

Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε superpixel των εικόνων που ανήκουν στο σύνολο του dataset εκπαίδευσης.

Έπειτα, και εφόσον έχουμε εξαγάγει τα χαρακτηριστικά SURF και Gabor, βρίσκουμε για το κάθε superpixel την μέση τιμή των descriptors και την μέση τιμή των διανυσμάτων για τα χαρακτηριστικά Gabor.

Τα αποτελέσματα αυτά ενώνονται όλα σε ένα διάνυσμα που αντιπροσωπεύει τα χαρακτηριστικά των superpixel και αποθηκεύονται στη λίστα **source_x**. Στη συνέχεια, αντιστοιχίζουμε τα διανύσματα με το αντίστοιχο χρώμα που υποδεικνύει το label, οι οποίες τιμές αποθηκεύονται στη λίστα **source_y**.

Επόμενο βήμα μας, πριν το τέλος της διαδικασίας, κανονικοποιούμε τις τιμές που έχουμε συλλέξει. Η κατασκευή του dataset για τις εικόνες εκπαίδευσης – πηγής υλοποιείται με την συνάρτηση **make_source_dataset**.

Αυτούσια η διαδικασία ακολουθείται και την εικόνα στόχου - target, με την βοήθεια της συνάρτησης **make_target_dataset**.

Αντίστοιχα, το διάνυσμα στο οποίο αποθηκεύονται τα μέσα διανύσματα για τα χαρακτηριστικά SURF και Gabor είναι το **target_x**.

```
Process: Creation of the target dataset.  
Completed: Creation of the target dataset.
```

Εικόνα Αποτελεσμάτων 13

Εκπαίδευση SVM Μοντέλου

Τη στιγμή κατά την οποία θα έχουμε έτοιμο το σύνολο εκπαίδευσης dataset με τα διανύσματα και τα labels, δημιουργούμε ένα Support Vector Machine (SVM) για την πρόβλεψη των χρωμάτων και στη συνέχεια τον χρωματισμό της εικόνας στόχου - target.

Για την ορθολογική υλοποίηση της συγκεκριμένης διαδικασίας δημιουργήσαμε τη συνάρτηση **svm_model_training**, η οποία δέχεται σαν ορίσματα τις λίστες που προαναφέρθηκαν: `source_x` και `source_y`.

Τέλος, αξίζει να σημειωθεί το γεγονός πως η εκπαίδευση του μοντέλου έγινε με τη βοήθεια της βιβλιοθήκης της Python, **sklearn** και συγκεκριμένα των συναρτήσεων της svm και predict.

Χρωματισμός της Εικόνας Στόχου.

Το τρίτο και τελευταίο βήμα για την επίλυση του 5^{ου} ζητούμενου της παρούσας εργασίας είναι ο χρωματισμός της εικόνας στόχου - target.

Πιο συγκεκριμένα, στο συγκεκριμένο βήμα θα περατωθεί με την βοήθεια της συνάρτησης **colorize_target**, η οποία δέχεται σαν ορίσματα τον ταξινομητή (Classifier) που προέκυψε από την εκπαίδευση του dataset για τις εικόνες εκπαίδευσης, τη λίστα `target_x` (δλδ. το dataset της εικόνας στόχος), την εικόνα στόχο - target και τα superpixel αυτής.

Στη συνέχεια και αφού τροφοδοτήσουμε το SVM μοντέλο με το σύνολο εκπαίδευσης της εικόνας target, αποθηκεύουμε τα id των χρωμάτων που προβλέφθηκαν.

Ειδικότερα, από τα συγκεκριμένα id, ενδιαφερόμαστε μόνο για τις τιμές A και B, τις οποίες και αποθηκεύουμε.

Έπειτα, δημιουργούμε μια κενή εικόνα με διαστάσεις ίσες της εικόνας στόχου - target, ενώ για κάθε superpixel αυτής χρωματίζουμε το κάθε εικονοστοιχείο - pixel σύμφωνα με την τιμή που προβλέφθηκε από το μοντέλο προηγουμένως.

Τελικό βήμα της παρούσας διαδικασίας είναι να εξάγουμε το τελικό αποτέλεσμα, για να γίνει αυτό μεταφερόμαστε από τον χρωματικό χώρο LAB στον RGB με την βοήθεια της μεθόδου:

```
cv2.cvtColor(colored_image, cv2.LAB2BGR)
```

Η μεταβλητή `colored_image` είναι η ζητούμενη χρωματισμένη εικόνα, με βάση την εκτίμησή που έκανε η υλοποίησή μας.

Αν όλα λειτουργήσουν ορθολογικά στο χρήστη θα εμφανιστεί το παρακάτω μήνυμα επιτυχίας:

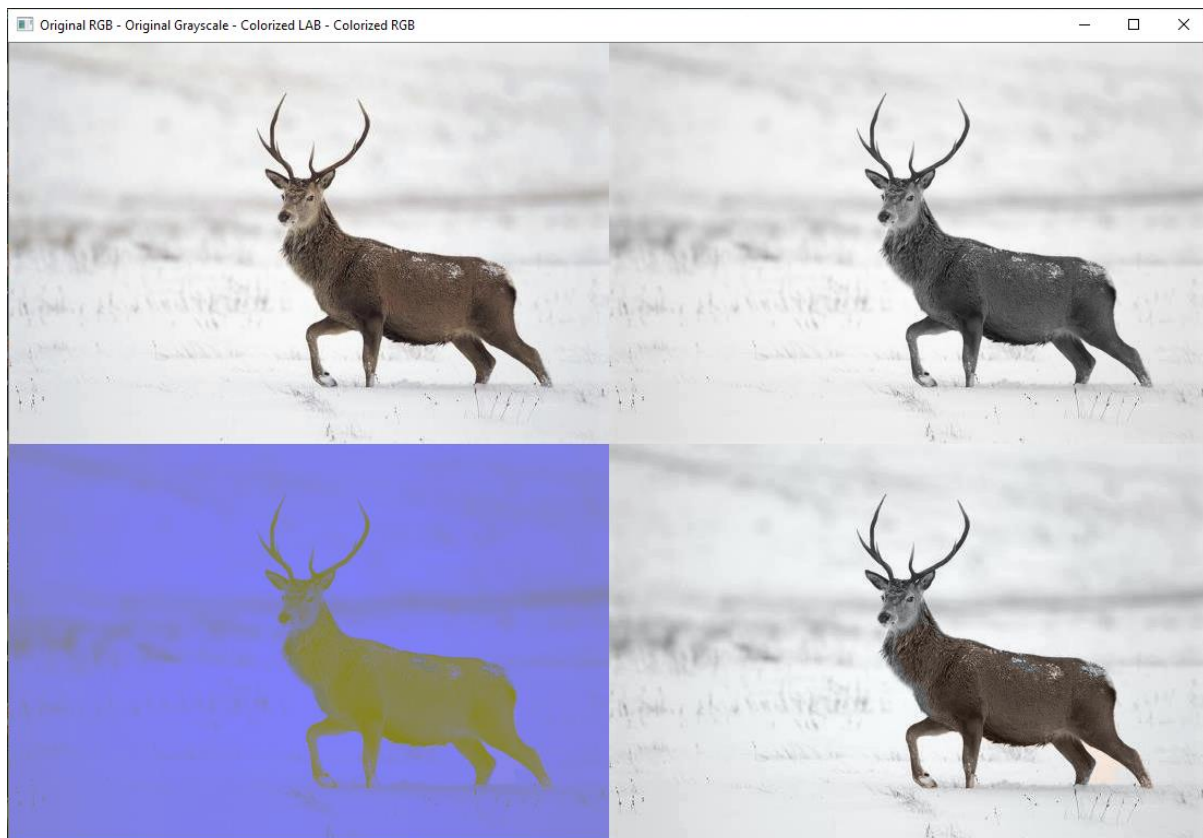
```
Process: Colorization of the target image.  
Completed: Colorization of the target image.
```

Εικόνα Αποτελεσμάτων 14

Αποτελέσματα Εκτέλεσης Υλοποίησης

Στη συγκεκριμένη θα παρουσιαστούν τα αποτελέσματα από την εκτέλεση της παρούσας υλοποίησης για τον χρωματισμό της εικόνας στόχου - target.

Για τα παρακάτω αποτελέσματα χρησιμοποιήθηκαν 5 εικόνες ως εικόνες του συνόλου εκπαίδευσης.



Εικόνα Αποτελεσμάτων 15

```
Process: Colorization of the target image.  
Completed: Colorization of the target image.
```

Εικόνα Αποτελεσμάτων 16

Οδηγίες Εκτέλεσης Προγράμματος

Στη συγκεκριμένη ενότητα θα αναφερθούμε στο τρόπο με τον οποίο θα εκτελεστεί ορθολογικά το εκτελέσιμο, που δημιουργήσαμε.

Ως εκ τούτου, για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

1. Ανοίγουμε τη γραμμή εντολών και μεταβαίνουμε στον αντίστοιχο φάκελο: \Ανάλυση Εικόνας.
2. Πληκτρολογούμε το όνομα του αρχείου main.py, ώστε να ανοίξει το κεντρικό αρχείο του εκτελέσιμου.

Το αρχείο εκτελείται επιτυχώς και τα αποτελέσματα εμφανίζονται όπως έχει προαναφερθεί σε παραπάνω ενότητα (βλ. Ενότητα: [Αποτελέσματα Εκτέλεσης Υλοποίησης](#)).

Βιβλιογραφία

Οι βιβλιογραφικές πηγές που χρησιμοποιήθηκαν στην εφαρμογή μας είναι οι ακόλουθες :

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”
http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/Colorization_main_6334%5b0%5d.pdf
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”
<https://people.cs.clemson.edu/~jzwang/ustc13/mm2012/p369-gupta.pdf>
7. <https://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
8. <https://cs.uwaterloo.ca/~zfrenett/CS886-Project.pdf>
9. http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/Colorization_main_6334%5b0%5d.pdf
10. <https://cvtuts.wordpress.com/2014/04/27/gabor-filters-a-practical-overview/>

Περιεχόμενα Απεσταλμένου Αρχείου

Το τελικό αρχείο της εργασίας (Εργασία Ανάλυσης Εικόνας.zip) θα περιέχει τα παρακάτω:

1. Τα αρχεία πηγαίου κώδικα σε γλώσσα Python που βρίσκονται στο φάκελο Ανάλυση Εικόνας.
2. Το κείμενο τεκμηρίωσης, με τίτλο Εργασία Ανάλυσης Εικόνας.pdf .
3. Το αρχείο συνοπτικής παρουσίασης, όπου θα εστιάζει στα κύρια σημεία της εργασίας, με τίτλο Παρουσίαση Ανάλυσης Εικόνας.ppsx.
4. Το αρχείο με τα στιγμιότυπα της εφαρμογής, που χρησιμοποιήθηκαν για την τεκμηρίωση της υλοποίησης, με τίτλο Screenshots.