

ΤΗΛ312 -1η Εργαστηριακή Άσκηση: Μοντέλα Χρωμάτων

Νίκου Γεώργιος-Νεκτάριος ΑΜ:2016030125
LAB31245385

1. Στο πρώτο ερώτημα μας ζητείται ένα πρόγραμμα που να διαβάζει έγχρωμες εικόνες, να τις μετατρέπει και να τις αποθηκεύει σε αποχρώσεις του γκρι. Αρχικά το πρόγραμμα ζητάει από τον χρήστη να δώσει όνομα αρχείου εικόνας για να εισαχθεί και μέσω `imread` η εικόνα διαβάζεται. Έπειτα η εικόνα μετατρέπεται σε grayscale μέσω της `rgb2gray` και αποθηκεύεται σε αρχείο `jpg` στον δίσκο με χρήση της `imwrite`.



Εικόνα 1: Grayscale εικόνα της fighter.jpg

Η grayscale εικόνα έχει μέγεθος 18kB, δηλαδή πάνω από δύο φορές μικρότερο μέγεθος από την αρχική. Αυτό συμβαίνει διότι η εικόνα μετατράπηκε από τις 3 διαστάσεις του πίνακα για την αποτύπωση των RGB χρωμάτων σε μονοδιάστατο πίνακα ο οποίος αποτυπώνει την εικόνα με παλέτα αποχρώσεων του γκρι.

2. Ο κωδικός της ομάδας εργασίας μου τελειώνει σε 385 επομένως το $k=7$. Αρχικά θέτω τις κατάλληλες τιμές στα `k`, `I`, `S`, `M` και `m` όπως υποδεικνύει η εκφώνηση. Έπειτα μέσω `for` loop γεμίζω πίνακα 256 θέσεων για το `H` με τιμές στο $(0,365)$ με βήμα $360/256$ για να πάρω 256 διαφορετικές τιμές του `H` μέσα σε αυτό το εύρος. Με την βοήθεια των τύπων της εκφώνησης, σε ένα βρόχο, βρίσκω για κάθε `πίξελ` τις RGB τιμές που αντιστοιχεί στο κάθε `H` και τις αποθηκεύω στον πίνακα RGB. Τέλος γεμίζω έναν πίνακα `new_image` με αριθμούς και μέσω της `imwrite` αποθηκεύουμε την εικόνα έχοντας ως ορίσματα τον πίνακα `new_image` ως `A` και τον πίνακα RGB ως `colormap`.



Εικόνα 2: Η παλέτα 256 χρωμάτων σε RGB μορφή

Τα αρχικά του μοντέλου HSI σημαίνουν: `H`=Hue δηλαδή το χρώμα που γίνεται αντιληπτό, `S`=Saturation, ο βαθμός καθαρότητας του χρώματος και `I`=Intensity, η ένταση φωτεινότητας της εικόνας. Το `H` είναι το

χρώμα που αντιλαμβανόμαστε, το S έχει να κάνει με το πόσο δυνατό ή αχνό είναι το εκάστοτε χρώμα και το I είναι η ένταση του φωτός της εικόνας.

3. Για το συγκεκριμένο ερώτημα το πρόγραμμα διαβάζει την εικόνα μέσω της `imread` και την εκτυπώνει με `imshow`. Μέσω της συνάρτησης `rgb2hsv` μετατρέπει την `rgb` εικόνα σε `hsv` μοντέλο και αποθηκεύει τις τιμές του πίνακα για το `saturation`. Έπειτα κάνει αφαιρεί από όλο τον πίνακα την τιμή 0.2 και ελέγχει όσες τιμές έγιναν αρνητικές να τις κάνει ίσες με το μηδέν. Έπειτα μετατρέπει ξανά την εικόνα σε `rgb` μορφή μέσω της `hsv2rgb` και την εμφανίζει. Η παραπάνω διαδικασία γίνεται άλλες 4 φορές καθώς το `saturation` παίρνει τιμές $[0,1]$ άρα μπορεί να γίνει συνολικά 5 φορές η αφαίρεση.



Εικόνα 3: Πάνω αριστερά η αρχική φωτογραφία και διαδοχικά κάθε φωτογραφία με `saturation` μικρότερο κατά 0.2

Το `saturation` έχει να κάνει με την ένταση ενός χρώματος και η μείωση του κάνει τα χρώματα να τείνουν προς το γκριζο.

4. Για το τελευταίο ερώτημα εισάγεται η εικόνα και γίνεται `resize` σε διαστάσεις $n \times n$, στην συγκεκριμένη περίπτωση 900×900 . Έπειτα αποθηκεύονται οι τιμές των `RGB` σε τρεις πίνακες. Για να βρεθούν οι μέσες τιμές των `R, G, B` γίνονται τρία διπλά `for loop` για τα διπλά αθροίσματα και εφαρμόζω τύποι της εκφώνησης. Έπειτα γίνονται οι συγκρίσεις μεταξύ των τιμών και εφόσον δεν είναι ίσες γίνονται διορθώσεις όπως ορίζουν οι τύποι και περνάμε τις νέες τιμές στην εικόνα. Τέλος προβάλλεται η εικόνα σε `subplot` μαζί με την αρχική και αποθηκεύεται.



Εικόνα 4: Αριστερά η αρχική εικόνα, Δεξιά η white-balanced εικόνα

Το white balance εξαρτάται από το είδος της πηγής του φωτός που έχει τραβηχτεί η φωτογραφία. Το αποτέλεσμα είναι αναμενόμενο διότι η συγκεκριμένη φωτογραφία έχει τραβηχτεί σε συνθήκες ηλιοφάνειας και με το πρόγραμμα έγιναν τα χρώματα πιο θερμά από ότι ήταν στην αρχική εικόνα.