

Ονοματεπώνυμο: Χαριτούδης Απόστολος-Ανδρέας

Μάθημα: Ανάλυση Εικόνας

Α.Μ.: π17178

Εξάμηνο: 7

Πανεπιστήμιο Πειραιά (ΠΑ.ΠΕΙ.)

Στις πρώτες γραμμές εισάγουμε τις βιβλιοθήκες και ταυτόχρονα εισάγουμε και την συνάρτηση `slic`, η οποία θα χρησιμοποιηθεί για να μετρήσουμε τα `superpixels`. Στην συνέχεια, ορίζουμε τις συναρτήσεις μας για τον υπολογισμό του χρώματος συγκεκριμένων σημείων της εικόνας. Το σημείο αυτό μπορεί να είναι οποιουδήποτε σχήματος.

Για την καθορισμένη περιοχή μάσκας μιας συγκεκριμένης εικόνας, η λειτουργία `segment_colorfulness` εκτελεί τις ακόλουθες εργασίες:

- Διαχωρίζει την εικόνα σε κανάλια συστατικών RGB,
- Καλύπτει την εικόνα χρησιμοποιώντας μάσκα, ώστε η χρωμάτωση να πραγματοποιείται μόνο στην καθορισμένη περιοχή, σε αυτήν την περίπτωση η περιοχή θα είναι το `superpixel` μας,
- Χρησιμοποιεί το `r` και `g` για τον υπολογισμό του `rg`,
- Χρησιμοποιεί τα στοιχεία RGB για τον υπολογισμό του `yb`,
- Υπολογίζει την μέση και τυπική απόκλιση των `yb` και `rg`, ενώ τα συνδυάζει,
- Τέλος, κάνει τον υπολογισμό της μέτρησης και επιστρέφει στην λειτουργία κλήσης.

Αφού ορίσαμε την συνάρτηση χρωμάτωσης, ώρα να ορίσουμε τις εντολές, με τις οποίες θα εισάγουμε την εικόνα.

- Με το `-image`, βάζουμε το `path` για την εισαγωγή της εικόνας και
- Με το `-segments`, ορίζουμε τον αριθμό των `superpixels` που θέλουμε.

Στην συνέχεια, φορτώνουμε την εικόνα στην μνήμη και μετράμε τα `superpixels`.

Φορτώνουμε την εντολή `-image` στην μνήμη ως `orig` και ακολουθούμε αυτό το βήμα ψάχνοντας ίδια σχήματα πλάτους και ύψους, όπως στο αρχικό `input` για την εικόνα μας. Έπειτα, ξαναφορτώνουμε την ίδια εντολή στην μνήμη, αλλά ως `image` σε `scikit-image` format. Ο λόγος που χρησιμοποιούμε αυτό το format είναι γιατί το

OpenCV φορτώνει τις εικόνες σε BGR κατάταξη, αντί σε RGB που το κάνει το scikit-image. Το slic, θα μετατρέψει την εικόνα μας σε L*a*b* color space κατά την διάρκεια της εύρεσης των superpixels υποθέτοντας ότι η εικόνα μας είναι σε RGB format. Οπότε έχουμε δύο τρόπους να φορτώσουμε την εικόνα. Και οι δύο τρόποι έχουν τα ίδια αποτελέσματα. Αφού υπολογιστούν τα superpixels, θα υπολογίσουμε την χρωμάτωση του κάθε pixel.

Πηγαίος Κώδικας:

```
# import the necessary packages
from skimage.exposure import rescale_intensity
from skimage.segmentation import slic
from skimage.util import img_as_float
from skimage import io
import numpy as np
import argparse
import cv2

def segment_colorfulness(image, mask):
    # split the image into its respective RGB components, then mask
    # each of the individual RGB channels so we can compute
    # statistics only for the masked region
    (B, G, R) = cv2.split(image.astype("float"))
    R = np.ma.masked_array(R, mask=mask)
    G = np.ma.masked_array(G, mask=mask)
    B = np.ma.masked_array(B, mask=mask)
    # compute rg = R - G
    rg = np.absolute(R - G)
    # compute yb = 0.5 * (R + G) - B
    yb = np.absolute(0.5 * (R + G) - B)
```

```

# compute the mean and standard deviation of both `rg` and `yb`,
# then combine them
stdRoot = np.sqrt((rg.std() ** 2) + (yb.std() ** 2))
meanRoot = np.sqrt((rg.mean() ** 2) + (yb.mean() ** 2))
# derive the "colorfulness" metric and return it
return stdRoot + (0.3 * meanRoot)

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
    help="path to input image")
ap.add_argument("-s", "--segments", type=int, default=100,
    help="# of superpixels")
args = vars(ap.parse_args())

# load the image in OpenCV format so we can draw on it later, then
# allocate memory for the superpixel colorfulness visualization
orig = cv2.imread(args["image"])
vis = np.zeros(orig.shape[:2], dtype="float")
# load the image and apply SLIC superpixel segmentation to it via
# scikit-image
image = io.imread(args["image"])
segments = slic(img_as_float(image), n_segments=args["segments"],
    slic_zero=True)

# loop over each of the unique superpixels
for v in np.unique(segments):
    # construct a mask for the segment so we can compute image
    # statistics for *only* the masked region
    mask = np.ones(image.shape[:2])
    mask[segments == v] = 0

```

```

# compute the superpixel colorfulness, then update the
# visualization array
C = segment_colorfulness(orig, mask)
vis[segments == v] = C

# scale the visualization image from an unrestricted floating point
# to unsigned 8-bit integer array so we can use it with OpenCV and
# display it to our screen
vis = rescale_intensity(vis, out_range=(0, 255)).astype("uint8")
# overlay the superpixel colorfulness visualization on the original
# image
alpha = 0.6
overlay = np.dstack([vis] * 3)
output = orig.copy()
cv2.addWeighted(overlay, alpha, output, 1 - alpha, 0, output)

# show the output images
cv2.imshow("Input", orig)
cv2.imshow("Visualization", vis)
cv2.imshow("Output", output)
cv2.waitKey(0)

```

Εντολή εκκίνησης:

```
$ AnalushEikonasSLIC.py --image (ονομαρχείου)
```