

Αναγνώριση Προτύπων

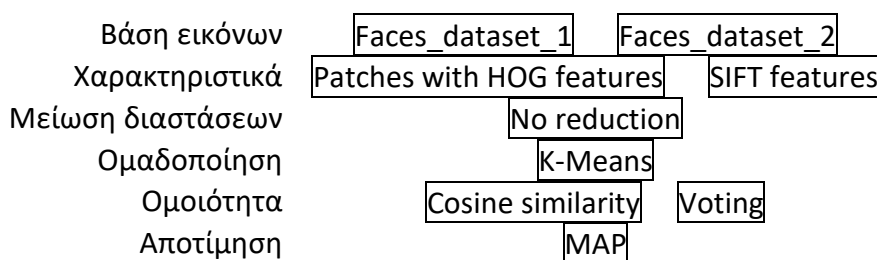
2^η ΕΡΓΑΣΙΑ

Θέμα: “Σύστημα ανάκτησης εικόνων”

Περιγραφή

Να γραφεί ένα πρόγραμμα σε Matlab/Octave το οποίο θα επιτρέπει την ανάκτηση δεδομένων από μια βάση εικόνων προσώπων με βάση το περιεχόμενό τους (Content Based Image Retrieval – CBIR). Δίνοντας μια εικόνα προσώπου ως ερώτημα (query) το πρόγραμμα θα πρέπει να επιστρέφει μια λίστα όλων των εικόνων της βάσης ταξινομημένα ως προς την ομοιότητά τους με την query εικόνα. Για τον σκοπό αυτό θα εξεταστούν διάφορες μέθοδοι για i) την εξαγωγή χαρακτηριστικών από τις εικόνες, ii) την ομαδοποίηση των χαρακτηριστικών στον χώρο των δεδομένων, iii) την εύρεση της ομοιότητας ανάμεσα στην αναζητούμενη εικόνα και τις εικόνες της βάσης, iv) την μείωση των διαστάσεων των χαρακτηριστικών. Η αποτίμηση της αποτελεσματικότητας αναζήτησης θα γίνει ποσοτικά με τον υπολογισμό του Mean Average Precision – MAP χρησιμοποιώντας όλες τις εικόνες της βάσης και ως queries.

Η βασική δομή της εφαρμογής μπορεί να περιγραφεί με τα βήματα που φαίνονται στο παρακάτω διάγραμμα:



Βάση εικόνων

Για τις πειραματικές δοκιμές θα χρησιμοποιηθούν δύο βάσεις προσώπων.

Faces_dataset_ATT (AT&T Database of faces) Περιέχει 10 εικόνες από 40 διαφορετικά πρόσωπα.

Faces_dataset_CLASS Περιέχει εικόνες προσώπων από την φωτογράφιση στα πλαίσια του μαθήματος.

Προτείνεται οι δοκιμές να γίνουν στην βάση `Faces_dataset_ATT` καθώς παρέχει ίδιο αριθμό εικόνων ανά πρόσωπο και όταν ολοκληρωθεί η εφαρμογή να τρέξουν πειράματα και για την `Faces_dataset_CLASS`.

Χαρακτηριστικά

Θα χρησιμοποιηθούν δύο προσεγγίσεις για την εξαγωγή των χαρακτηριστικών.

i) **Patches** Κάθε εικόνα της βάσης διαιρείται σε ένα σύνολο μη επικαλυπτόμενων patches. Για κάθε patch υπολογίζονται κάποια χαρακτηριστικά οπότε στη συνέχεια το patch εκφράζεται ως ένα διάνυσμα με τις τιμές των χαρακτηριστικών. Για το σκοπό αυτό θα χρησιμοποιηθούν τα Histogram of Oriented Gradients - HOG features. Συνεπώς, η εικόνα εκφράζεται σαν ένας πίνακας $N \times 81$, όπου N το πλήθος των patches και 81 το μήκος του κάθε HOG feature vector.

ii) **Keypoints** Για κάθε εικόνα της βάσης εξάγονται τα SIFT features της. Συνεπώς, η εικόνα εκφράζεται σαν ένας πίνακας χαρακτηριστικών $N \times 128$ όπου N το πλήθος των keypoints που βρέθηκαν και 128 το μήκος του κάθε SIFT feature vector.

Στο αρχείο `hog_features_demo.m` που σας δίνεται υπάρχει η υλοποίηση της εξαγωγής των HOG χαρακτηριστικών από patches της εικόνας.

Στο αρχείο `sift_features_demo.m` που σας δίνεται υπάρχει η υλοποίηση της εξαγωγής των SIFT χαρακτηριστικών της εικόνας.

Οδηγίες

- Διαβάστε προσεκτικά τα **σχόλια** στα αρχεία `hog_features_demo.m` και `sift_features_demo.m`. Η δομή τους είναι παρόμοια όμως διαφέρουν στον τρόπο υπολογισμού των χαρακτηριστικών.
- Τα αρχεία τρέχουν σε Matlab και Octave.
- Για τα HOG features χρησιμοποιείται και το αρχείο `hog.m` που κάνει τον σχετικό υπολογισμό.
- Για τα SIFT features χρησιμοποιείται και το πρόγραμμα `siftWin32.exe` (original κώδικας του D. Lowe).
- Και στα δύο αρχεία, εάν βγάλετε τα σχόλια στις γραμμές 30-34 τότε δεν γίνεται υπολογισμός των αντίστοιχων δεδομένων αλλά διαβάζονται απευθείας από τα αρχεία. Αυτό επιταχύνει την εκτέλεση του συνολικού προγράμματος ώστε να μην υπολογίζονται τα χαρακτηριστικά κάθε φορά που το τρέχετε (ενώ πλέον θέλετε να εστιάσετε σε παρακάτω μέρη του προγράμματος, π.χ. K-Means). Πάντως, τουλάχιστον μια φορά θα πρέπει προφανώς να υπολογιστούν ώστε να δημιουργηθούν τα αντίστοιχα `.mat` αρχεία.
- Και στα δύο αρχεία η γραμμή 48 είναι προσωρινή ώστε να υπολογίζονται μόνο οι πρώτες 3 εικόνες της βάσης. Έχει μείνει ώστε να υπάρξει μια αρχική αίσθηση για την ταχύτητα εκτέλεσης. Κάντε την σχόλιο ώστε ο υπολογισμός να γίνει για όλες τις εικόνες.
- Ο υπολογισμός των HOG features είναι σημαντικά πιο αργός στο Octave.

Μείωση διαστάσεων

Τα πειράματα θα γίνουν χωρίς να εφαρμοστεί κάποια τεχνική μείωσης διαστάσεων των χαρακτηριστικών.

Ομαδοποίηση

Θα χρησιμοποιηθεί ο **K-Means** αλγόριθμος για τον υπολογισμό του συνόλου των οπτικών λέξεων (Bag of Visual Words – BoVW).

Στο αρχείο `kmeans_demo.m` που σας δίνεται υπάρχει η υλοποίηση του αλγόριθμου K-Means για τον υπολογισμό των κέντρων των κλάσεων.

Οδηγίες

- Διαβάστε προσεκτικά τα **σχόλια** στο αρχείο `kmeans_demo.m`.
- Το αρχείο τρέχει σε Matlab και Octave.
- Όπως και προηγουμένως, εάν βγάλετε τα σχόλια στις γραμμές 29-33 τότε δεν γίνεται υπολογισμός του K-Means αλλά διαβάζονται τα δεδομένα ταξινόμησης απευθείας από τα αρχεία.
- Εάν έχετε υπολογίσει τα HOG features (`hog_features_demo.m`) και έχετε υπολογίσει και την ομαδοποίηση σε BoVW (`kmeans_demo.m`) τότε μπορείτε να τρέξετε το `show_patches_demo.m` για να δείτε τα αποτελέσματα. Διαβάστε προσεκτικά τα σχόλια στον κώδικα του αρχείου αυτού. Δείχνει πώς αντιστοιχίζονται τα patches σε visual words. Επίσης, εμφανίζει ένα ιστόγραμμα με την συχνότητα εμφάνισης των visual words.
- Εάν έχετε υπολογίσει τα SIFT features (`sift_features_demo.m`) και έχετε υπολογίσει και την ομαδοποίηση σε BoVW (`kmeans_demo.m`) τότε μπορείτε να τρέξετε το `show_points_demo.m` για να δείτε τα αποτελέσματα. Διαβάστε προσεκτικά τα σχόλια στον κώδικα του αρχείου αυτού. Δείχνει πώς αντιστοιχίζονται τα keypoints σε visual words. Επίσης, εμφανίζει ένα ιστόγραμμα με την συχνότητα εμφάνισης των visual words.

Ομοιότητα

Έχοντας υπολογίσει τα χαρακτηριστικά για τις εικόνες της βάσης είτε με HOG είτε με SIFT και έχοντας και την αντιστοίχιση των features σε visual words, κάθε εικόνα εκφράζεται πλέον ως ένα διάνυσμα N οπτικών λέξεων όπου N το πλήθος είτε των patches είτε των keypoints, αντίστοιχα.

Έστω μια εικόνα προς αναζήτηση (**query**). Για αυτήν υπολογίζονται τα features της (είτε HOG είτε SIFT) και έπειτα τα features της αντιστοιχίζονται σε visual words από την υπάρχουσα ομαδοποίηση που προέκυψε προηγουμένως από τον KMeans για τις εικόνες της βάσης. Συνεπώς, και η query εικόνα εκφράζεται ως ένα διάνυσμα N οπτικών λέξεων.

Στην συνέχεια, υπολογίζεται η ομοιότητα μεταξύ της query εικόνας και όλων των εικόνων της βάσης. Εν τέλει, προκύπτει ένα score (ομοιότητας) για κάθε εικόνα της βάσης.

Ο υπολογισμός της ομοιότητας των εικόνων της βάσης προς την query εικόνα θα βασιστεί σε δύο μεθόδους:

i) Cosine Similarity, όπως αναφέρεται στις σελίδες 17 και 32 των σημειώσεων **Lec03 Ανάκτηση εικόνων.pdf**. Τιμές κοντά στη μονάδα αντιστοιχούν σε μεγάλη ομοιότητα.

ii) Voting process, όπως περιγράφεται στην μέθοδο των Nister and Stewenius **nister_stewenius_cvpr2006_slides.ppt** σελ. 33-36. Στην παρούσα εργασία, δεν θα χρησιμοποιηθεί ιεραρχική δομή αλλά τα απευθείας αποτελέσματα της ομαδοποίησης που έχει προκύψει προηγουμένως από τον KMeans.

Στο αρχείο `voting_demo.m` που σας δίνεται υπάρχει η υλοποίηση του παραδείγματος της παρουσίασης των Nister and Stewenius, σελ 33-36.

Οδηγίες

- Διαβάστε προσεκτικά τα **σχόλια** στο αρχείο `voting_demo.m`.
- Το αρχείο τρέχει σε Matlab και Octave.
- Προσαρμόστε το στα δεδομένα της εργασίας ώστε να υπολογίζει το score για κάθε εικόνα της βάσης.

Αποτίμηση

Οι εικόνες της βάσης ταξινομούνται κατά φθίνουσα σειρά με βάση τα scores τους ως προς την query εικόνα, όπως φαίνεται στο παράδειγμα στην σελίδα 41 των σημειώσεων **Lec03 Ανάκτηση εικόνων.pdf** και υπολογίζεται το **Average Precision (AP)** για την αυτή την query εικόνα. Το **ground truth**, δηλαδή οι relevant εικόνες της βάσης που πραγματικά αντιστοιχούν στην query εικόνα δίνονται από το filename του query. Για παράδειγμα, εάν η query εικόνα είναι η 4^η εικόνα του 2^{ου} προσώπου, δηλαδή η `s02_04.pgm` τότε οι relevant εικόνες (ground truth) είναι όλες οι `s02_*.pgm`. Προφανώς, η ίδια η query εικόνα θα είναι πρώτη στην κατάταξη των scores. Το ζητούμενο είναι πόσο «ψηλά» στην λίστα θα βρίσκονται οι υπόλοιπες 9 εικόνες του συγκεκριμένου προσώπου. Αυτό απεικονίζεται στην τιμή του AP.

Στο αρχείο `map_demo.m` που σας δίνεται υπάρχει η υλοποίηση του παραδείγματος σελίδα 41 των σημειώσεων σε 2 παραλλαγές.

Οδηγίες

- Διαβάστε προσεκτικά τα **σχόλια** στο αρχείο `map_demo.m`.
- Το αρχείο τρέχει σε Matlab και Octave.
- Προσαρμόστε το στα δεδομένα της εργασίας ώστε να υπολογίζεται το Average Precision για κάποια query εικόνα.

Πειραματικά αποτελέσματα

i) Για μια query εικόνα



Θεωρήστε ως query την 1^η εικόνα του 10^{ου} προσώπου, δηλαδή την `s10_01.pgm`.

Τρέξτε το πρόγραμμα που υλοποιήσατε και υπολογίστε το **Average Precision (AP)** για τα εξής σενάρια:

	HOG Patches No reduction Cosine similarity	HOG Patches No reduction Voting	SIFT features No reduction Cosine similarity	SIFT features No reduction Voting
KMeans K=20				
KMeans K=100				

ΕΡΩΤΗΣΗ 1

Για κάθε πείραμα της πρώτης γραμμής του παραπάνω πίνακα (KMeans K=20):

- Αποθηκεύστε το αρχείο Matlab που το «τρέχει» σε ξεχωριστό filename. Συνεπώς, δημιουργήστε κατά σειρά τα αρχεία:





















`k20_hog_cosine.m`

k20_hog_voting.m
k20_sift_cosine.m
k20_sift_voting.m

- Συμπληρώστε σε μια στήλη ενός πίνακα του Word to Top-5 των αποτελεσμάτων ανάκτησης. Στην τελευταία γραμμή δώστε το Average Precision (για όλες τις εικόνες της βάσης).

Οδηγίες

- Ένα υπόδειγμα (με τυχαία αποτελέσματα) σας δίνεται παρακάτω. Προφανώς, σε αυτό το παράδειγμα, το δεύτερο setup (HOG Patches - No reduction – Voting) δίνει καλύτερα αποτελέσματα καθώς έχει σωστά 4 από τα 5 πρώτα αποτελέσματα.

HOG Patches No reduction Cosine similarity	HOG Patches No reduction Voting	SIFT features No reduction Cosine similarity	SIFT features No reduction Voting
 OK	 OK	 False	 OK
 False	 False	 OK	 False
 False	 OK	 False	 False
 OK	 OK	 False	 OK
 False	 OK	 OK	 False
AP= ...	AP= ...	AP= ...	AP= ...

ii) Για το σύνολο της βάσης

Θεωρήστε ως query **όλες** τις εικόνες της βάσης. Τρέξτε το πρόγραμμα που υλοποιήσατε και υπολογίστε το **Mean Average Precision (MAP)** για τα εξής σενάρια:

	HOG Patches No reduction Cosine similarity	HOG Patches No reduction Voting	SIFT features No reduction Cosine similarity	SIFT features No reduction Voting
KMeans K=20				
KMeans K=100				

ΕΡΩΤΗΣΗ 2

Αποθηκεύστε το αρχείο που τρέχει το 1^ο setup, δηλαδή, το K20 – HOG – Cosine similarity σε ένα αρχείο

`final_k20_hog_cosine.m`

Σχολιάστε τα αποτελέσματα του παραπάνω πίνακα. Ποια συμπεράσματα προκύπτουν ως προς την σύγκριση HOG – SIFT και ποια ως προς την σύγκριση Cosine similarity – Voting; Πόσο επηρεάζει το πλήθος K των κλάσεων τα αποτελέσματα; Ποια τεχνική ωφελείται περισσότερο από την αύξηση του K και ποια λιγότερο; Προτείνετε κάποια ερμηνεία για αυτή την διαφορά.

ΕΡΩΤΗΣΗ 3

Τελικά ποιο είναι το καλύτερο MAP που μπορείτε να πετύχετε; Με ποιο setup το πετυχαίνετε;

ΕΡΩΤΗΣΗ 4

Δοκιμάστε το καλύτερο setup σας της Ερώτησης 3 στην δεύτερη βάση των εικόνων από την τάξη. Τι συνολικό MAP πετυχαίνετε;

Παράδοση εργασίας

Το κείμενο **.docx** της εργασίας θα πρέπει αναφέρει τα στοιχεία σας και να περιέχει μια αναλυτική παρουσίαση της εργασίας και των παραπάνω ερωτημάτων.

Παραδοτέο: ένα συμπιεσμένο αρχείο **ergasia2.zip** (ή **ergasia2.rar**) που να περιέχει τα αρχεία

- **k20_hog_cosine.m**
- **k20_hog_voting.m**
- **k20_sift_cosine.m**
- **k20_sift_voting.m**
- **final_k20_hog_cosine.m**
- **ergasia2.docx**

και αποστολή ΜΟΝΟ μέσω e-class στην ενότητα Εργασίες μέχρι και την **Κυριακή 26/1/2020** στις **12:00 το βράδυ**.

Για οποιαδήποτε διευκρίνιση μπορείτε να επικοινωνήσετε μαζί μου στο **akesidis@uniwa.gr** ή να αποστείλετε μήνυμα στο eclass, στην ενότητα «**Συζητήσεις**» του μαθήματος.

Τάσος Κεσίδης