



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΤΟΜΕΑΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ Η/Υ & ΒΙΟΜΗΧ. ΕΦΑΡΜΟΓΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΑΞΙΝΟΜΗΣΗ ΨΗΦΙΑΚΩΝ ΕΙΚΟΝΩΝ  
ΜΕ ΧΩΡΙΚΗ-ΣΗΜΑΣΙΟΛΟΓΙΚΗ ΑΝΤΙΣΤΟΙΧΗΣΗ

**ΣΠΟΥΔΑΣΤΗΣ : ΦΩΤΙΑΔΗΣ ΚΥΡΙΑΚΟΣ**  
**A.M. 1151**

ΕΠΙΒΛΕΠΩΝ : Δρ. ΝΙΚΟΛΑΙΔΗΣ ΑΘΑΝΑΣΙΟΣ  
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

ΣΕΡΡΕΣ, ΦΕΒΡΟΥΑΡΙΟΣ 2015



## ΠΕΡΙΛΗΨΗ

Η αναγνώριση του περιεχόμενου μίας εικόνας και η ταξινόμηση της εικόνας σε κατηγορίες είναι ένα σημαντικό πρόβλημα στους κλάδους της Αναγνώρισης Προτύπων και Τεχνητής Νοημοσύνης. Η επίλυση του προβλήματος αυτού μπορεί να φέρει σημαντικά αποτελέσματα και να χρησιμοποιηθεί σε πολλές εφαρμογές.

Κατά καιρούς έχουν αναπτυχθεί διάφορες μέθοδοι ταξινόμησης εικόνας, όπως είναι ο σάκος οπτικών λέξεων-BoVW(bags of visual word) η οποία είναι μία μέθοδος που βασίζεται στην αναπαράσταση μιας εικόνας με τη χρήση τοπικών χαρακτηριστικών, Η ταξινόμηση των εικόνων γίνεται σύμφωνα με αυτά τα χαρακτηριστικά. Μια άλλη μέθοδος είναι η χωρική ταξινόμηση πυραμίδας-SPM(spatial pyramid matching),κατά την οποία η εικόνα χωρίζεται σε υποπεριοχές, κάθε διάνυσμα που προκύπτει από μια υποπεριοχή είναι ένα κομμάτι της τελικής αναπαράστασης της εικόνας, και ευθυγραμμίζοντας τις υποπεριοχές αυτές με τον ίδιο τρόπο θα πρέπει να πάρουμε δυο παρόμοια διανύσματα εάν οι δύο εικόνες μοιάζουν.

Στη παρούσα πτυχιακή εργασία μελετάται και υλοποιείται η μέθοδος σημασιολογικής-χωρικής ταξινόμησης-SSM(semantic-spatial-matching), η οποία λαμβάνει υπόψη όχι μόνο τη χωρική αλλά και την σημασιολογική απεικόνιση. Για την σημασιολογική απεικόνιση εξάγονται τα χαρακτηριστικά της κάθε εικόνας όπως αυτή χρησιμοποιήθηκε στη μέθοδο BoVW, και για την χωρική ταξινόμηση υιοθετούμε την τεχνική που αναπτύχθηκε στη μέθοδο SPM. Το τελικό αποτέλεσμα προκύπτει συνδυάζοντας αυτές τις δύο μεθόδους.

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή.....	6
1.ΠΑΛΑΙΟΤΕΡΟΙ ΑΛΓΟΡΙΘΜΟΙ ΤΑΞΙΝΟΜΗΣΗΣ.....	8
1.1 Εισαγωγή	
1.2BoVW (bags of visual words).....	9
1.3 SPM(spatial pyramid matching).....	10
2.ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....	13
2.1 k-Means.....	13
2.2 SIFT.....	16
2.3 SVM(support vector machine).....	18
3. SSM (semantic-spatial matching).....	22
3.1 Εισαγωγή.....	22
3.2 Σημασιολογικός χώρος.....	22
3.3 Σημασιολογικό ταίριασμα.....	23
3.4 SSM (semantic-spatial matching).....	24
4 Σύνοψη.....	25
ΠΑΡΑΡΤΗΜΑ Α.....	28
ΠΑΡΑΡΤΗΜΑ Β.....	43

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

σχ.1.1SPM Διαίρεση εικόνας σε περιοχές και απεικόνιση των διανυσμάτων χαρακτηριστικών.....	11
σχ.1.2SPM Δυο σχεδόν ίδιες εικόνες με διαφορετική διάταξη των αντικειμένων στο χώρο ,παράγουν δύο τελείως διαφορετικά ιστογράμματα σύμφωνα με την μέθοδο SPM.....	12
σχ.2.1 k-means Αρχικοποίησηcentroid.....	14
σχ.2.2 k-means Ομαδοποίηση δεδομένων.....	14
σχ.2.3 k-means Επανατοποθέτησηcentroid.....	14
σχ.2.4 k-means Εκ νέου ομαδοποίηση δεδομένων.....	15
σχ.2.5 k-means Τελική μορφή.....	15
σχ.2.6svm διαχωριστική γραμμή, απόσταση margin, support vectors.....	19
σχ.2.7svm επικαλυπτόμενες κλάσεις.....	19
σχ.2.8svm γραμμικά μη διαχωριζόμενες κλάσεις.....	20
Πίνακας 4.1.....	25
Εικ. 15scene dataset.....	26
Πίνακας 4.2.....	26
Πίνακας 4.3.....	27

## ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια ο όγκος των ψηφιακών εικόνων που έχουμε να διαχειριστούμε έχει αυξηθεί απότομα. Αυτό οφείλεται στο χαμηλό κόστος των ψηφιακών φωτογραφικών μηχανών, αλλά και στο ότι κάθε κινητό τηλέφωνο διαθέτει και ένα ικανοποιητικής ευκρίνειας φωτογραφικό φακό που του δίνει τη δυνατότητα να λειτουργεί σαν φωτογραφική μηχανή. Έτσι είναι πάρα πολύ εύκολη η λήψη φωτογραφιών ανά πάσα στιγμή κατά τη διάρκεια της μέρας, όπως επίσης η αποθήκευση και η ανάκτησή τους. Παράλληλα, η δημοτικότητα των social media που μία από τις κύριες λειτουργίες τους είναι η προβολή εικόνων έχει προκαλέσει μια τεράστια μεταφορά δεδομένων από και προς το διαδίκτυο. Επίσης εκτός από ψυχαγωγικούς λόγους οι άνθρωποι βρίσκονται καθημερινά αντιμέτωποι με προβλήματα ταξινόμησης εικόνων σε πολλούς επιστημονικούς κλάδους.

Ο ανθρώπινος εγκέφαλος όταν λαμβάνει μια εικόνα παράγει αυτόματα κάποιες πληροφορίες για την εικόνα αυτή, όπως για παράδειγμα ποιο είναι το θέμα της εικόνας(βουνό, πάρτι, αυτοκινητόδρομος), τι αντικείμενα περιέχει η εικόνα (αεροπλάνο, φλιτζάνι, τραπέζι), την αναγνώριση κάποιων συγκεκριμένων αντικειμένων (ο φίλος μου ο Νίκος, το παλιό μου αυτοκίνητο), το οποίο είναι ένα αρκετά δύσκολο ζήτημα για τις μηχανές. Με το δεδομένο ότι οι βάσεις δεδομένων που διαχειριζόμαστε πλέον είναι τεράστιες, η ταξινόμηση τους είναι μια επίπονη και χρονοβόρα διαδικασία που είναι σχεδόν αδύνατο ένας άνθρωπος να ταξινομεί καθημερινά χιλιάδες εικόνες μόνος του βλέποντάς τες μια-μια. Υπάρχει λοιπόν η ανάγκη να αναπτυχθούν οι κατάλληλοι μηχανισμοί που πραγματοποιούν όλες αυτές τις λειτουργίες. Ένα τέτοιο επίτευγμα θα δώσει λύση σε πολλά προβλήματα, θα μειώσει το κόστος, θα διευκολύνει την καθημερινότητα των ανθρώπων και θα χρησιμοποιηθεί σε διάφορες εφαρμογές. Μερικά παραδείγματα είναι:

- Οι κάμερες ασφαλείας θα μπορούν να αναγνωρίζουν κάποια ασυνήθιστη δραστηριότητα χωρίς την εποπτεία ανθρώπων.
- Θα μπορούν να χρησιμοποιούνται ρομπότ που είναι ικανά να λαμβάνουν αποφάσεις σε καταστάσεις επικίνδυνες για τον άνθρωπο.

- Θα μπορεί κάποιος να βρει εύκολα και γρήγορα φωτογραφίες με συγκεκριμένα κοινά χαρακτηριστικά.

Στο παρελθόν έχουν γίνει διάφορες προσπάθειες δημιουργίας τέτοιων μηχανισμών με ικανοποιητικά αποτελέσματα αλλά και με πολλά περιθώρια βελτίωσης. Στα επόμενα κεφάλαια θα γίνει αναφορά σε αλγόριθμους που χρησιμοποιούνται στις μεθόδους ταξινόμησης εικόνας, επίσης θα γίνει αναφορά σε παλαιότερες μεθόδους ταξινόμησης. Έπειτα στο κύριο μέρος της εργασίας θα αναφερθούν τα βήματα υλοποίησης όπως και ο κώδικας του αλγορίθμου που εξετάζεται (SSM-semantic-spatial matching) .

## 1. ΠΑΛΑΙΟΤΕΡΟΙ ΑΛΓΟΡΙΘΜΟΙ ΤΑΞΙΝΟΜΗΣΗΣ

### 1.1 Εισαγωγή

Η μέθοδος SSM δεν δημιουργήθηκε αμέσως, είναι η εξέλιξη άλλων προγενέστερων μεθόδων. Οι πρόγονοι της SSM είναι η BoVW και η SPM. Η μέθοδος BoVW (bag of visual word) γνώρισε μεγάλη επιτυχία στην ταξινόμηση εικόνων. Η μέθοδος βασίζεται στην εξαγωγή τοπικών χαρακτηριστικών μέσω του αλγόριθμου SIFT (scale invariant feature transform). Αρχικά όλα τα τοπικά χαρακτηριστικά ενός σετ εικόνων εξάγονται και κατηγοριοποιούνται μέσω του αλγορίθμου κ-μέσων (k-means), το αποτέλεσμα των κ-μέσων είναι το codebook, κάθε καταχώρηση του codebook είναι μία οπτική λέξη (codeword), έπειτα κάθε χαρακτηριστικό μιας εικόνας ελέγχεται με ποια οπτική λέξη βρίσκεται πιο κοντά και έτσι δημιουργείται ένα ιστόγραμμα χαρακτηριστικών για κάθε εικόνα. Ως επέκταση της μεθόδου BoVW επινοήθηκε η μέθοδος SPM (spatial pyramid matching). Η SPM στηρίζεται στη χωρική απεικόνιση της εικόνας, για να το πετύχει αυτό χωρίζει την εικόνα σε περιοχές και εξάγει τα τοπικά χαρακτηριστικά για κάθε υποπεριοχή, η εικόνα αναπαρίσταται ενώνοντας όλα τα ιστογράμματα των υπό-περιοχών σε ένα μακρύ τελικό διάνυσμα. Η μέθοδος SM σε αντίθεση με την SPM στηρίζεται στην σημασιολογική απεικόνιση της εικόνας και όχι στη χωρική. Το σημείο κλειδί είναι η κατασκευή ενός σημασιολογικού χώρου. Για να επιτευχθεί αυτό τα ιστογράμματα χαρακτηριστικών όλων των εικόνων και των υπό-περιοχών τους κατηγοριοποιούνται με τον αλγόριθμο κ-μέσων και έτσι προκύπτει ο σημασιολογικός χώρος. Έπειτα κάθε ιστόγραμμα περιοχής ελέγχεται με ποιο ιστόγραμμα του σημασιολογικού χώρου βρίσκεται πιο κοντά και παίρνει την εκάστοτε ετικέτα. Βάση των ετικετών τους τα ιστογράμματα μιας εικόνας ευθυγραμμίζονται. Ο συνδυασμός της SPM και SM είναι η μέθοδος SSM. Τέλος για την ταξινόμηση των εικόνων χρησιμοποιούνται μηχανισμοί διανυσμάτων υποστήριξης(SVM).



## 1.2BoVW (bags of visual words)

Τα τελευταία χρόνια η εικονική αναπαράσταση μιας εικόνας παίζει πολύ σημαντικό ρόλο για την ταξινόμηση της. Πάνω σε αυτό βασίζεται η μέθοδος ταξινόμησης BoVW. Η συγκεκριμένη μέθοδος έχει δείξει ευρωστία σε διάφορα backgrounds ή σε περιπτώσεις επικαλύψεις αντικειμένων μέσα σε μια εικόνα.

Η εκτέλεση αυτού του αλγορίθμου βασίζεται στην εξαγωγή τοπικών χαρακτηριστικών. Η διαδικασία εκτέλεσης της μεθόδου BoVW έχει ως εξής:

- Αρχικά από ένα σετ εικόνων εξάγονται τα τοπικά χαρακτηριστικά, όλα τα χαρακτηριστικά από όλες τις εικόνες τοποθετούνται σε ένα χώρο(σάκο οπτικών λέξεων).
- Έπειτα μέσα από ένα αλγόριθμο ομαδοποίησης(π.χ. k-means) δημιουργούμε k αντιπροσωπευτικούς περιγραφείς (codewords). Οι τοπικοί περιγραφείς αποτελούν το λεξικό οπτικών λέξεων (codebook). Το μέγεθος του codebook είναι ο αριθμός k στον οποίο χωρίσαμε το πλήθος όλων των περιγραφών.
- Στη συνέχεια για κάθε περιοχή εξάγονται οι τοπικοί περιγραφείς και καθένας από αυτούς ελέγχεται με ποιον από τους αντιπροσωπευτικούς περιγραφείς βρίσκεται πιο κοντά. Η απόσταση μεταξύ ενός τοπικού περιγραφέα και ενός αντιπροσωπευτικού περιγραφέα μετριέται με Ευκλείδεια Απόσταση.
- Τέλος δημιουργείται ένα ιστόγραμμα χαρακτηριστικών για κάθε εικόνα το οποίο έχει μήκος ίσο με το codebook και δείχνει πόσοι τοπικοί περιγραφείς της εικόνας ταιριάζουν με κάθε οπτική λέξη (codeword).

Παρόλο που ο αλγόριθμος BoVW έχει γνωρίσει μεγάλη επιτυχία και έχει χρησιμοποιηθεί σε πολλές εφαρμογές έχει ένα σημαντικό μειονέκτημα, δεν

λαμβάνει καθόλου υπόψη την χωρική αναπαράσταση της εικόνας, δηλαδή ταιριάζει τις εικόνες σύμφωνα με τα κοινά τοπικά χαρακτηριστικά χωρίς να ενδιαφέρεται για το σε ποιο σημείο της εικόνας βρίσκονται τα χαρακτηριστικά και έτσι αγνοεί το καθολικό θέμα της εικόνας.

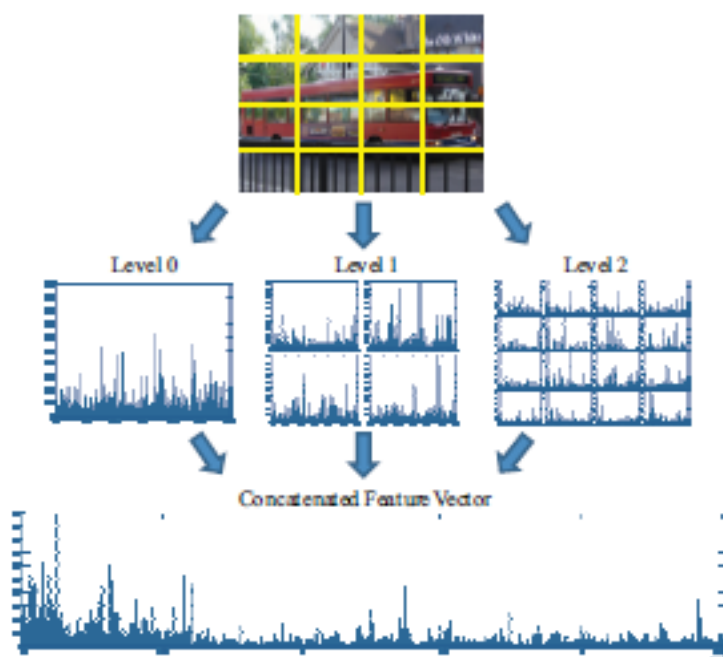
Για την αντιμετώπιση αυτού του προβλήματος έχουν αναπτυχθεί διάφοροι αλγόριθμοι, αυτός που αναφέρεται στη συνέχεια ως μία εξέλιξη του BoVW είναι ο SPM.

### **1.3 SPM(spatial pyramid matching)**

Το πρόβλημα που επιχειρεί να αντιμετωπίσει η μέθοδος SPM(spatial pyramid matching, χωρικό ταίριασμα πυραμίδας) είναι η καθολική αναπαράσταση μιας εικόνας χωρίς να χρειάζεται να αναγνωρίσει πρώτα τα επιμέρους αντικείμενα που βρίσκονται στην εικόνα αυτή. Για παράδειγμα σε μια εικόνα που απεικονίζει μία παραλία δεν χρειάζεται πρώτα να αναγνωρίσει θάλασσα, ήλιο, αμμουδιά, σέρφερ, καρχαρία κ.τ.λ. για να την τοποθετήσει στη σωστή κατηγορία. Η μέθοδος είναι ιδιαίτερα αποδοτική σε εικόνες που περιέχουν μία σκηνή(δάση, βουνά κ.τ.λ.).

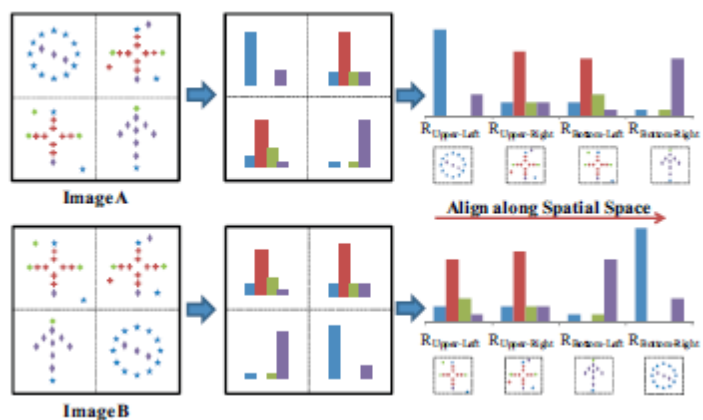
Η μέθοδος SPM είναι μια επέκταση της μεθόδου B.o.V.W(bags of visual words) η οποία όμως δίνει ιδιαίτερη βαρύτητα στη χωρική απεικόνιση του περιεχομένου μιας εικόνας. Η τεχνική αυτής της μεθόδου βασίζεται στην διαίρεση της εικόνας σε υπό-περιοχές και αντιμετωπίζει την κάθε υπό-περιοχή όπως αυτή γινόταν στη μέθοδο B.o.V.W. Έπειτα όλα τα ιστογράμματα που προκύπτουν από κάθε υπό-περιοχή ευθυγραμμίζονται σε ένα διάνυσμα για την αναπαράσταση ολόκληρης της εικόνας. Πιο συγκεκριμένα ξεκινώντας από το πρώτο επίπεδο(level 0) χρησιμοποιείται ολόκληρη η εικόνα παίρνοντας το ιστόγραμμα χαρακτηριστικών (ίδιο με B.o.V.W.). Στο επόμενο επίπεδο (level 1) η εικόνα τεμαχίζεται σε τέσσερα κομμάτια, διαιρώντας κάθε πλευρά της εικόνας στη μέση, και έτσι παράγεται ένα ιστόγραμμα χαρακτηριστικών για κάθε υπό-περιοχή. Με τον ίδιο τρόπο κάθε υπό-περιοχή διαιρείται στα τέσσερα, έτσι στο επόμενο επίπεδο (level 2) η εικόνα είναι χωρισμένοι σε δεκαέξι υπό-περιοχές για τις οποίες έχουν δημιουργηθεί

ιστογράμματα χαρακτηριστικών που ευθυγραμμίζονται με προκαθορισμένο τρόπο σε ένα μακρύ διάνυσμα το οποίο αντιστοιχεί σε ολόκληρη την εικόνα. Ο αριθμός των υπό-περιοχών σε κάθε επίπεδο ισούται με  $4^L$  όπου L το επίπεδο της διαίρεσης. Το μήκος του τελικού διανύσματος εξαρτάται από το μέχρι πιο επίπεδο λειτουργεί ο αλγόριθμος και από το μήκος των διανυσμάτων που παράγονται. Η τεχνική με την οποία τεμαχίζεται μια εικόνα αναπαριστάται στο παρακάτω σχήμα.



σχ.1.1SPM Διαίρεση εικόνας σε περιοχές και απεικόνιση των διανυσμάτων χαρακτηριστικών

Η μέθοδος SPM βασίζεται στην υπόθεση ότι δύο εικόνες από την ίδια κλάση θα έχουν πανομοιότυπο χωρικά περιεχόμενο. Δηλαδή η πάνω αριστερή γωνία της μίας εικόνας θα πρέπει να έχει ίδιο οπτικό περιεχόμενο με την πάνω αριστερή γωνία της άλλης εικόνας. Αυτό μπορεί να λειτουργεί σε κάποιες περιπτώσεις αλλά ένα αντικείμενο μπορεί να βρίσκεται οπουδήποτε μέσα σε μια εικόνα. Έτσι αν δύο εικόνες ανήκουν στην ίδια κλάση και απλά έχουν τα κοινά τους αντικείμενα διαφορετικά κατανομημένα στο χώρο, η μέθοδος SPM αποτυγχάνει να κάνει το σωστό ταίριασμα. Αυτό φαίνεται εύκολα στο παρακάτω σχήμα όπου οι δύο εικόνες είναι σχεδόν ίδιες και το μόνο που διαφέρει είναι η διάταξη των αντικειμένων στο χώρο, έτσι τα ιστογράμματα που προκύπτουν δεν μοιάζουν καθόλου μεταξύ τους.



σχ.1.2SPM Δυο σχεδόν ίδιες εικόνες με διαφορετική διάταξη των αντικειμένων στο χώρο, παράγουν δύο τελείως διαφορετικά ιστογράμματα σύμφωνα με την μέθοδο SPM

Για να αντιμετωπιστεί αυτό το πρόβλημα θα πρέπει ο τρόπος που παρασκευάζεται το τελικό διάνυσμα να έχει εκτός από χωρικά και σημασιολογικά κριτήρια. Για το λόγο αυτό προτάθηκε ο αλγόριθμος SSM(semantic-spatial matching), που βασίζεται τόσο στη χωρική όσο και στη σημασιολογική ταξινόμηση μιας εικόνας.

## 2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

### 2.1 k-Means

Ο αλγόριθμος k-Means είναι ένας δημοφιλής αλγόριθμος κατηγοριοποίησης και χρησιμοποιείται ευρέως στο κλάδο της Αναγνώρισης Προτύπων. Σκοπός του αλγορίθμου είναι να χωρίσει τα δεδομένα σε k ομάδες. Ός είσοδο ο αλγόριθμος παίρνει ένα σύνολο παρατηρήσεων και τον αριθμό των k κεντροειδών (centroids), δηλαδή τον αριθμό των ομάδων στις οποίες θέλουμε να χωρίσουμε τα δεδομένα. Έξοδος του συστήματος είναι τα δεδομένα εισόδου με ένα label που δείχνει σε ποια ομάδα ανήκουν, και ένα αντιπροσωπευτικό διάνυσμα για κάθε κέντρο ομάδας(centroid).

Η φιλοσοφία του αλγορίθμου είναι να δημιουργήσει k ομάδες για τις οποίες τα μέλη της κάθε ομάδας θα απέχουν μεταξύ τους την ελάχιστη δυνατή απόσταση, και τα μέλη διαφορετικών ομάδων θα απέχουν μεταξύ τους την μέγιστη δυνατή απόσταση. Η απόσταση μεταξύ των σημείων αλλά και μεταξύ των σημείων και των κεντροειδών μπορεί να υπολογιστεί με διάφορους τρόπους με τον συνηθέστερο να είναι η Ευκλείδεια Απόσταση.

Για την υλοποίηση του αλγορίθμου αρχικά προσδιορίζεται η θέση των centroid, η οποία συνήθως γίνεται με τυχαίο τρόπο. Αξίζει να σημειωθεί ότι τα αποτελέσματα εξαρτώνται από την αρχική θέση των centroid , έτσι για διαφορετικά αρχικά centroid ο αλγόριθμος δίνει διαφορετικά αποτελέσματα, οπότε είναι προτιμότερο ο αλγόριθμος k-Means να εφαρμόζεται αρκετές φορές για να πάρουμε το βέλτιστο δυνατό αποτέλεσμα. Στο επόμενο βήμα το καθένα από τα δεδομένα συσχετίζεται με το centroid που βρίσκεται πιο κοντά σε αυτό, έτσι δημιουργούνται οι ομάδες στην αρχική τους μορφή. Έπειτα για να γίνει καλύτερος διαμερισμός των παρατηρήσεων επαναπροσδιορίζονται τα centroids παίρνοντας θέση στο κέντρο της ομάδας τους. Έχοντας νέα centroid επαναλαμβάνεται η διαδικασία εύρεσης του κοντινότερου centroid για κάθε παρατήρηση. Έτσι με αυτή τη διαδικασία είναι πιθανό κάποιες παρατηρήσεις να αλλάξουν ομάδα στην οποία ανήκουν. Οι δύο αυτές διαδικασίες επαναλαμβάνονται έως ότου να μην υπάρχουν μετακινήσεις παρατηρήσεων μεταξύ των ομάδων και άρα τα centroid να μένουν σταθερά. Το

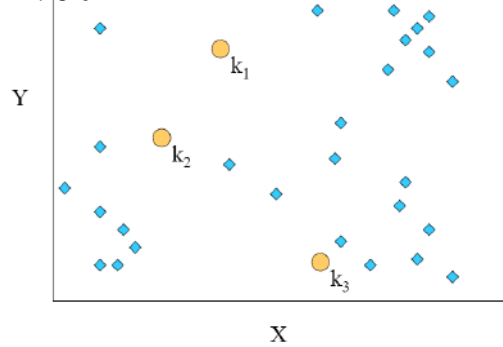
αποτέλεσμα είναι η τιμή που έχει το κάθε centroid και οι παρατηρήσεις που ανήκουν σε κάθε centroid.

- Βημα1: Τυχαία αρχικοποίηση κέντρων.

Αρχική κατάσταση.

$K = 3$  συστάδες

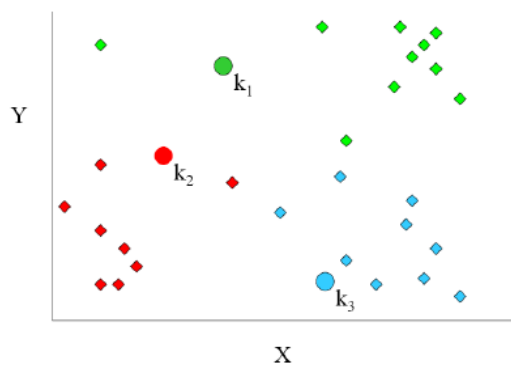
Αρχικά σημεία  $k_1, k_2, k_3$



σχ.2.1 k-means Αρχικοποίηση centroid

- Βήμα2: Εύρεση του αριθμού των παρατηρήσεων που ανήκουν σε κάθε κέντρο.

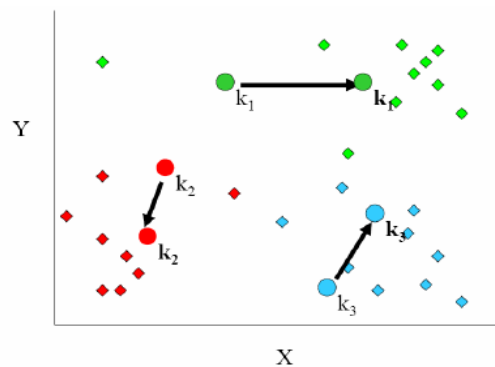
Τα σημεία ανατίθενται στο πιο γειτονικό από τα 3 αρχικά σημεία



σχ.2.2 k-means Ομαδοποίηση δεδομένων

- Βημα3: Επανατοποθέτηση των κέντρων στο κέντρο βάρους της κάθε ομάδας.

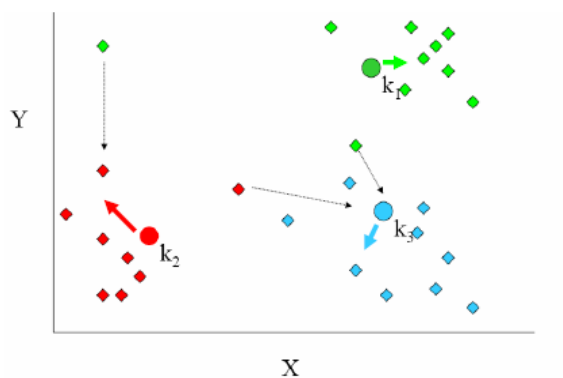
Επανα-υπολογισμός του κέντρου (κέντρου βάρους) κάθε σημείου



σχ.2.3 k-means Επανατοποθέτηση centroid

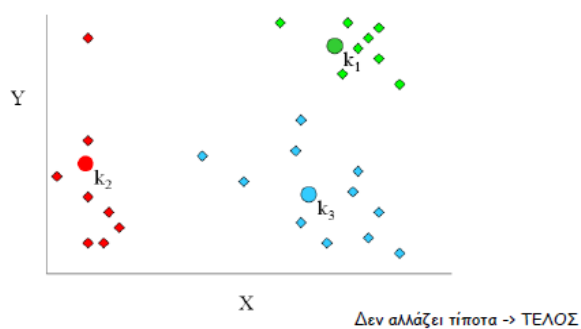
Βήμα4:Επανάληψη του βήματος 2 και 3 μέχρι να μην υπάρχουν άλλες μετακινήσεις.

Νέα ανάθεση των σημείων  
Νέα κέντρα βάρους



σχ.2.4 k-means Εκ νέου ομαδοποίηση δεδομένων

- Βήμα5:Τελικά αποτελέσματα κατηγοριοποίησης του k-Means.



σχ.2.5 k-means Τελική μορφή

## 2.2 SIFT

Ένα αναπόσπαστο κομμάτι στο ταίριασμα εικόνων είναι η ανάλυση και επεξεργασία της πληροφορίας που περιέχεται μέσα στην εικόνα. Για την επεξεργασία του περιεχομένου μιας εικόνας χρησιμοποιούνται αλγόριθμοι εξαγωγής χαρακτηριστικών. Κατά καιρούς έχουν μελετηθεί και αναπτυχθεί διάφοροι αλγόριθμοι για αυτήν τη διαδικασία, μερικοί από τους οποίους είναι οι αλγόριθμοι SURF(Speeded Up Robust Features), RIFT(Rotation Invariant Feature Transform) και SIFT(Scale Invariant Feature Transform). Σε αυτήν την εργασία χρησιμοποιείται ο αλγόριθμος SIFT, ο οποίος είναι αρκετά αποτελεσματικός και χρησιμοποιείται ευρέως για την εξαγωγή χαρακτηριστικών.

Τα τοπικά χαρακτηριστικά που εξάγονται με τον αλγόριθμο από μια εικόνα πρέπει να πληρούν κάποιες προϋποθέσεις, ώστε να είναι κατάλληλα για να ταιριάξουν σωστά ένα αντικείμενο που βρίσκεται σε δύο διαφορετικές εικόνες. Για να είναι κατάλληλο ένα χαρακτηριστικό θα πρέπει να:

1. Είναι αμετάβλητο στη κλίμακα. Δηλαδή ανεξάρτητα αν ένα αντικείμενο μπορεί να καλύπτει ολόκληρη την εικόνα ή να βρίσκεται σε μία άκρη αυτής, θα πρέπει να αναγνωρίζεται από τον αλγόριθμο.
2. Μένει αμετάβλητο στο φόντο της εικόνας. Δηλαδή θα πρέπει να αναγνωρίζεται ανεξάρτητα από το τι υπάρχει πίσω ή γύρω από αυτό.
3. Μένει ανεπηρέαστο στο θόρυβο. Πολλές φορές σε μια εικόνα ένα αντικείμενο να μη φαίνεται ολόκληρο ή να καλύπτεται ένα μέρος του από κάτι άλλο.
4. Να αναγνωρίζεται ανεξάρτητα από ποιά οπτική γωνία βρίσκεται σε κάθε εικόνα.
5. Τέλος ένα αντικείμενο να αναγνωρίζεται όταν αποτυπώνεται κάτω από διαφορετικό φωτισμό.

Ένα τοπικό χαρακτηριστικό SIFT αναγνωρίζεται από τρεις συντελεστές: α) θέση(location) β) κλίμακα (scale) γ) προσανατολισμό (orientation) και περιγράφεται



από ένα διάνυσμα μήκους 128(descriptor).

Για τον εντοπισμό τοπικών χαρακτηριστικών(SIFT) από μία εικόνα χρησιμοποιείται μια σειρά από φίλτρα. Τα κύρια στάδια για τον υπολογισμό είναι:

1. Ανίχνευση ακρότατων στο χώρο και στην κλίμακα(scale-space extrema detection): Στο πρώτο στάδιο εφαρμόζοντας την συνάρτηση Gauss ερευνούνται όλες οι πιθανές θέσεις που παραμένουν ανεπηρέαστες στην κλιμάκωση και στην περιστροφή.
2. Εντοπισμός σημείων κλειδιών (key point localization): Σε κάθε υποψήφια θέση καθορίζεται η θέση και η κλίμακα εφαρμόζοντας ένα λεπτομερές μοντέλο . Τα σημεία κλειδιά επιλέγονται με βάση τη σταθερότητα.
3. Ανάθεση προσανατολισμού(orientation assignment): Ένας ή περισσότεροι προσανατολισμοί ανατίθενται σε κάθε πιθανό σημείο κλειδί με βάση τις τοπικές διευθύνσεις κλίσης της εικόνας . Όλες οι μελλοντικές ενέργειες εκτελούνται σε κάθε χαρακτηριστικό, έτσι όπως έχει μετασχηματιστεί σύμφωνα με τον προσανατολισμό, την κλίμακα και τη θέση του. Με αυτό τον τρόπο οι μετασχηματισμοί αυτοί παρέχουν σταθερότητα.
4. Περιγραφείς των σημείων κλειδιών(key point descriptor): Υπολογίζεται η κλίση της εικόνας στην κλίμακα που επιλέχτηκε γύρο από κάθε σημείο κλειδί. Αυτή μεταμορφώνεται σε αναπαράσταση που επιτρέπει την παραμόρφωση σχημάτων και αλλαγή φωτισμού σε υψηλά επίπεδα.

Με αυτή τη διαδικασία δημιουργείται ένας μεγάλος αριθμός τοπικών χαρακτηριστικών. Αυτό παίζει σημαντικό ρόλο, γιατί σε αντικείμενα που είναι δύσκολα αναγνωρίσιμα απαιτούνται περισσότερα χαρακτηριστικά.

## 2.3 SVM(support vector machine)

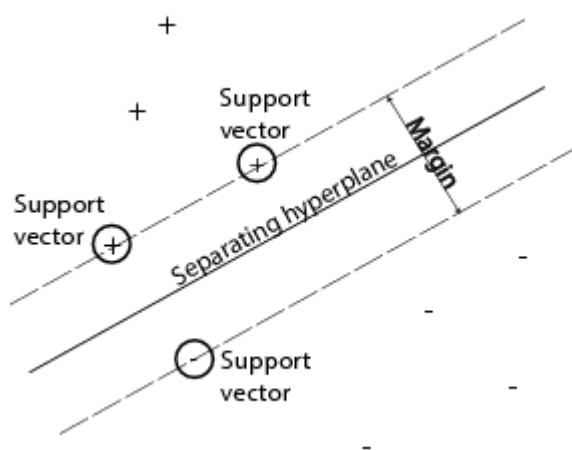
Οι μηχανισμοί διανυσμάτων υποστήριξης-svm(support vector machine) είναι δημοφιλείς αλγόριθμοι ταξινόμησης. Οι αλγόριθμοι ταξινόμησης είναι αλγόριθμοι εκπαίδευσης, δηλαδή επεξεργάζονται τα δεδομένα και προσπαθούν να βρουν μέσα από αυτά κάποια σχέση που τα συνδέει μεταξύ τους, ώστε να εντοπίσουν κάποια μοτίβα ή κανόνες. Σύμφωνα με αυτούς τους κανόνες ο αλγόριθμος εκπαιδεύεται και έτσι "αποφασίζει" για την συμπεριφορά των δεδομένων που θα έρθουν στη συνέχεια για ταξινόμηση.

Οι αλγόριθμοι svm είναι δυαδικοί αλγόριθμοι ταξινόμησης, δηλαδή λαμβάνουν δεδομένα και τα χωρίζουν σε δυο κατηγορίες. Για να γίνει αποτελεσματικά αυτός ο διαχωρισμός θα πρέπει να έχουμε τα κατάλληλα δεδομένα. Κατάλληλα δεδομένα σημαίνει ότι στην αναπαράσταση τους στο χώρο θα πρέπει να είναι σαφής ο διαχωρισμός τους.

Τα δεδομένα που δίνονται για εκπαίδευση είναι ένα σύνολο παρατηρήσεων και οι κλάσεις στις οποίες ανήκει η κάθε παρατήρηση. Ο αλγόριθμος στο στάδιο της εκπαίδευσης προσπαθεί να ανακαλύψει τη συνάρτηση που συνδέει κάθε παρατήρηση με την αντίστοιχη κλάση της, ώστε να μπορέσει να βρει ένα γενικό κανόνα. Έτσι, όταν θα λάβει παρατηρήσεις για ταξινόμηση χρησιμοποιώντας τον κανόνα που προέκυψε από την εκπαίδευση, να αποδίδει την σωστή κλάση.

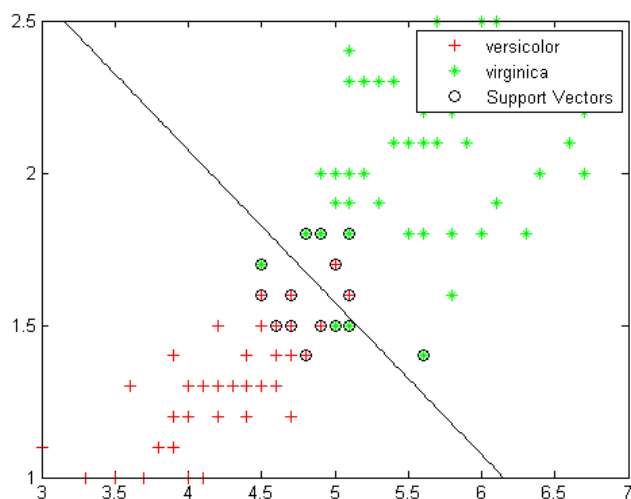
Βασική προϋπόθεση για την λειτουργία του αλγορίθμου είναι τα δεδομένα να είναι γραμμικά διαχωριζόμενα στο χώρο. Δηλαδή να εντοπίζεται μια ευθεία η οποία να χωρίζει τις δυο κλάσεις. Η ευθεία για να είναι βέλτιστη θα πρέπει να έχει όσο το δυνατόν μεγαλύτερη απόσταση από τα κοντινότερα σημεία των δύο κλάσεων. Η απόσταση ανάμεσα στα σημεία των δυο κλάσεων που είναι πιο κοντά μεταξύ τους ονομάζεται απόσταση margin. Είναι προφανές ότι όσο πιο μεγάλη είναι η απόσταση margin τόσο πιο αποτελεσματικός είναι ο αλγόριθμος. Οι παρατηρήσεις που οριοθετούν την αρχή της κάθε κλάσης λέγονται διανύσματα υποστήριξης (support

vectors). Αυτά φαίνονται καλύτερα στο παρακάτω σχήμα.



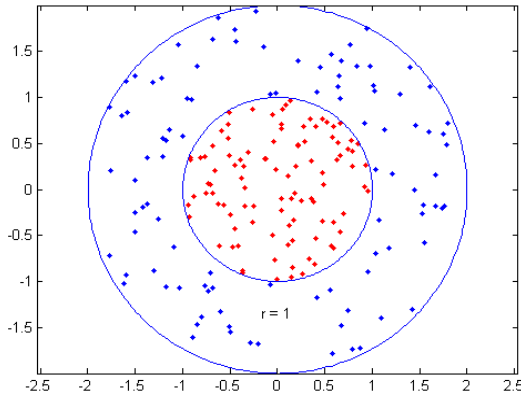
σχ.2.6svm διαχωριστική γραμμή, απόσταση margin, support vectors

Σε πραγματικές συνθήκες την περίπτωση τα δεδομένα μας να είναι γραμμικά διαχωριζόμενα δεν την συναντάμε πολύ συχνά. Κάποιες φορές τα δεδομένα των δυο κλάσεων επικαλύπτονται, σε αυτές τις περιπτώσεις τα σφάλματα είναι αναπόφευκτα και αναμενόμενα. Κατά συνέπεια η επιλογή της ευθείας που τα διαχωρίζει πρέπει είναι τέτοια ώστε να έχει το μικρότερο ποσοστό λαθών.



σχ.2.7svm επικαλυπτόμενες κλάσεις

Επιπλέον συναντάται συχνά τα δεδομένα να βρίσκονται το ένα μέσα στο άλλο, όπως φαίνεται παρακάτω.



σχ.2.8svm γραμμικά μη διαχωριζόμενες κλάσεις

Σε αυτήν την περίπτωση σκοπός είναι να απεικονιστούν τα δεδομένα σε ένα χώρο περισσότερων διαστάσεων έτσι ώστε να βρεθεί ένα υπέρ-επίπεδο που να μπορεί να χωρίζει γραμμικά τα δεδομένα. Με αυτόν τον τρόπο λύνεται το πρόβλημα των μη γραμμικών διαχωριζόμενων κλάσεων και ο αλγόριθμος είναι σε θέση να κάνει το διαχωρισμό. Οι συναρτήσεις που μεταφέρουν τα δεδομένα από ένα χώρο σε έναν άλλο διαφορετικής διάστασης ονομάζονται συναρτήσεις πυρήνα(kernel function).

Οι πιο διάσημες συναρτήσεις πυρήνα είναι οι:

- Γραμμικού πυρήνα (linear kernel)
- Πολυώνυμου πυρήνα (polynomial)
- Συνάρτηση Ακτινικής Βάσης (Radial base Function)
- Τετραγωνικός πυρήνας (quadratic kernel)

Οι αλγόριθμοι svm παρόλο που είναι δυαδικοί ταξινομητές, με χρήση πολλών svm θα μπορούσαν να λειτουργήσουν και ως ταξινομητές πολλών κλάσεων. Αυτό μπορεί να συμβεί με δυο περιπτώσεις:

- i. Η πρώτη είναι η περίπτωση ένας εναντίον όλων, όπου κάθε κλάση διαχωρίζεται απέναντι σε όλες τις άλλες. Όποτε  $M$  δυαδικοί ταξινομητές απαιτούνται για  $M$  κλάσεις. Το όριο της κάθε κλάσης έρχεται μετά από συμψηφισμό των  $M$  svm ταξινομητών.
- ii. Η δεύτερη είναι η περίπτωση ένας εναντίον ενός, όπου κάθε κλάση χρησιμοποιεί έναν ταξινομητή για κάθε μία από όλες τις άλλες κλάσεις. Άρα

για αυτή την περίπτωση απαιτούνται  $M*(M-1)/2$  δυαδικοί ταξινομητές. Έτσι σε κάθε παρατήρηση για κάθε ταξινομητή  $s_{nm}$  που συμμετέχει αποδίδεται μια κλάση. Η παρατήρηση παίρνει την κλάση που της έχει αποδοθεί τις περισσότερες φορές.

### 3. SSM (semantic-spatial matching)

#### 3.1 Εισαγωγή

Η μέθοδος SPM διαιρεί μια εικόνα σε υποπεριοχές. Το πρόβλημα όμως είναι πως η μέθοδος αδυνατεί να κάνει σωστό ταίριασμα ανάμεσα σε δύο υποπεριοχές δύο διαφορετικών εικόνων. Μια πιθανή λύση θα ήταν να συγκρίνονται όλες οι υποπεριοχές από όλες τις εικόνες μία προς μία, για να βρεθεί το καλύτερο ταίριασμα. Αυτή όμως η προσέγγιση έχει κάποια βασικά μειονεκτήματα. Πρώτα από όλα ένας τέτοιος αλγόριθμος θα ήταν χρονοβόρος, θα απαιτούσε πολύ υπολογιστική ισχύ και θα ήταν δύσκολο να δημιουργηθεί προγραμματιστικά. Δεύτερον δε θα υπάρχει ένας προκαθορισμένος οδηγός που να υλοποιεί το τελικό ιστόγραμμα. Έτσι τα διανύσματα των εικόνων δεν θα έχουν ίδιο μήκος ιστογραμμάτων και δεν υπάρχει δυνατότητα για επέκταση και σε άλλες εφαρμογές. Και τέλος δεν είναι απόλυτο ότι μια υποπεριοχή θα ταιριάζει με μία άλλη, είναι πιθανό να μην ταιριάζει με καμία ή να ταιριάζει με πολλές άλλες. Για να λυθούν αυτά τα προβλήματα εισάγεται μία νέα μέθοδος αντιστοίχισης, η μέθοδος SM (semantic matching).

#### 3.2 Σημασιολογικός χώρος

Για να υλοποιηθεί αυτή η μέθοδος χρειάζεται να φτιαχτεί ένας ενοποιημένος σημασιολογικός χώρος. Για να κατορθώσουμε αυτό τοποθετούμε όλα τα ιστογράμματα περιοχών των εικόνων σε έναν αλγόριθμο κατηγοριοποίησης όπου το κάθε κέντρο που προκύπτει από την κατηγοριοποίηση αντιπροσωπεύει ένα σημασιολογικό υποχώρο. Το σύνολο των υπό-χώρων αντιπροσωπεύει το σημασιολογικό χώρο. Πιο συγκεκριμένα, για ένα σετ εικόνων χωρίζουμε τις εικόνες σε υπό-περιοχές όπως ακριβώς και στη μέθοδο SPM. Θεωρούμε  $R = \{r_1, r_2, \dots, r_N\}$  όπου  $R$  είναι ο σημασιολογικός χώρος και  $\{r_1, r_2, \dots, r_N\}$  είναι τα διανύσματα χαρακτηριστικών από όλες τις υποπεριοχές όλων των εικόνων. Τοποθετώντας όλα αυτά τα διανύσματα στον αλγόριθμο k-Means έχουμε κατασκευάσει τον σημασιολογικό χώρο λύνοντας τη παρακάτω εξίσωση:

$$\min_S = \sum_{i=1}^N (\min_{k=1, \dots, k} \|r_i - r_k\|^2)$$

όπου  $S = \{s_1, s_2, \dots, s_k\}$  είναι τα κέντρα που προκύπτουν από το k-Means, αντιπροσωπεύουν ένα σημασιολογικό υποχώρο και το καθένα αναπαριστά μία ετικέτα (semantic label).

### 3.3 Σημασιολογικό ταίριασμα

Έχοντας κατασκευάσει το σημασιολογικό χώρο, πλέον είναι πιο εύκολο να διαχειριστούμε ένα ταίριασμα μεταξύ δύο υποπεριοχών δυο διαφορετικών εικόνων. Κάθε ιστόγραμμα χαρακτηριστικών μιας υποπεριοχής παίρνει μία ετικέτα (semantic label) σύμφωνα με τον αντιπροσωπευτικό σημασιολογικό υποχώρο που βρίσκεται πιο κοντά. Την απόσταση ανάμεσα σε δυο διανύσματα τη μετράμε βάση της Ευκλείδειας Απόστασης. Με αυτό τον τρόπο μπορεί να δημιουργηθεί ένας προκαθορισμένος οδηγός που να υλοποιεί το τελικό ιστόγραμμα, βάζοντας όλα τα διανύσματα υποπεριοχών σε μια σειρά σύμφωνα με την ετικέτα τους.

Πιο συγκεκριμένα μια εικόνα  $I$  διαιρείται σε  $M$  περιοχές,  $R = \{r_1, r_2, \dots, r_M\}$  όπου το  $r_i$  είναι το διάνυσμα χαρακτηριστικών της  $i$  υποπεριοχής. Κάθε  $r_i$  παίρνει μία ετικέτα ανάλογα με το ποιο διάνυσμα βρίσκεται πιο κοντά στο σημασιολογικό χώρο  $S$ . Έπειτα ευθυγραμμίζονται όλα τα διανύσματα χαρακτηριστικών σύμφωνα με την ετικέτα τους ( $s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_m$ ). Όσα ιστογράμματα έχουν ίδια ετικέτα προστίθενται και το τελικό διάνυσμα μιας εικόνας δίνεται από την σχέση:

$$V_I(SM) = [r_{s_1}^T, r_{s_2}^T, \dots, r_{s_m}^T]^T$$

Όπου  $r_{s_k}$  το άθροισμα όλων των ιστογραμμάτων που έχουν την  $k$  ετικέτα. Αν κανένα διάνυσμα δεν έχει ταιριάζει με μια ετικέτα, τότε η ετικέτα προστίθεται στο τελικό διάνυσμα αποτελούμενη με μηδενικά.

Ο αλγόριθμος SM είναι αποδοτικός και ευέλικτος καθώς θέτοντας διαφορετικό αριθμό ετικετών μπορούμε να δημιουργούμε έναν αυστηρό ή έναν χαλαρό αλγόριθμο ανάλογα με τις ανάγκες των δεδομένων. Αν θέσουμε ένα μικρό αριθμό

ετικετών  $k$  τότε ο αλγόριθμος θα έχει αδυναμία να διακρίνει δύο αντικείμενα που μοιάζουν αλλά είναι διαφορετικά και θα τους αποδίδει λανθασμένα την ίδια σημασιολογική ετικέτα. Αντίθετα, αν θέσουμε έναν μεγάλο αριθμό ετικετών, ο αλγόριθμος θα γίνει πολύ αυστηρός στο διαχωρισμό των ιστογραμμάτων και ιστογράμματα που ανήκουν στην ίδια σημασιολογική ομάδα και θα έπρεπε να έχουν την ίδια ετικέτα δεν θα ταιριάζουν. Αυτό καθιστά τον αλγόριθμο μη ανθεκτικό στο θόρυβο.

### 3.4 SSM (semantic-spatial matching)

Η μέθοδος SPM (spatial pyramid matching) χωρίζει την εικόνα σε περιοχές και με έναν προκαθορισμένο τρόπο συνδέει τα ιστογράμματα χαρακτηριστικών των περιοχών σε ένα μακρύ διάνυσμα, για την αναπαράσταση της εικόνας. Η μέθοδος SM (semantic matching) συνδέει τις περιοχές μιας εικόνας σύμφωνα με την ετικέτα που έχει πάρει η κάθε μία μέσα από ένα σύνολο ετικετών.

Για να συνδέσουμε τα διανύσματα από αυτές τις δυο μεθόδους υιοθετούμε τη μέθοδο ένωσης γραμμικού πυρήνα

$$SSM = a * SPM + (1-a) * SM$$

Όπου  $a$  είναι ένας συντελεστής με πεδίο τιμών  $[0-1]$  και δείχνει την επιρροή της κάθε μεθόδου στο τελικό διάνυσμα.



## 4 ΣΥΝΟΨΗ

Για την κατανόηση της συμπεριφοράς του αλγορίθμου έγιναν μερικά πειράματα αλλάζοντας τις τιμές των παραμέτρων σε διάφορα σημεία του αλγορίθμου. Τα συμπεράσματα που προκύπτουν φαίνονται παρακάτω.

Αρχικά θα θέλαμε να παρατηρήσουμε πώς αντιδρά ο αλγόριθμος σε διαφορετικές τιμές του codebook και τον αριθμό των σημασιολογικών ετικετών.

Πίνακας 4.1

ετικέτες codebook	180	240	300	360	420
200	82%	85%	87%	85%	83%
400	88%	89%	95%	85%	84%

Όπως φαίνεται από τον πίνακα ο αλγόριθμος αποδίδει καλύτερα με ένα δυνατό διάλυσμα χαρακτηριστικών  $N=400$  (μεγάλο codebook) παρά με ένα πιο αδύναμο  $N=200$ . Επίσης η ακρίβεια ποικίλλει ανάλογα με τον αριθμό των σημασιολογικών ετικετών που θα επιλέξουμε. Για το συγκεκριμένο παράδειγμα  $C=300$  φαίνεται να είναι η αποδοτικότερη επιλογή. Στο συγκεκριμένο παράδειγμα χρησιμοποιήθηκαν 5 από τις 15 κλάσεις του dataset, συγκεκριμένα οι κλάσεις: opencountry, store, street, suburb, tallbuilding.



Εικ. 15scene dataset

Γενικά η ταξινόμηση πολλών κλάσεων με τη χρήση μηχανών διανυσμάτων υποστήριξης δεν είναι ένα απλό θέμα και πολλές παράμετροι πρέπει να ληφθούν υπόψη. Κύριο ρόλο παίζει η σωστή επιλογή των παραμέτρων συναρτήσεων πυρήνα, και ο αριθμός των κλάσεων που επηρεάζει την ακρίβεια στο αποτέλεσμα του ταξινομητή, καθώς όσες περισσότερες κλάσεις υπάρχουν προς ταξινόμηση τόσο δυσκολεύεται ο αλγόριθμος να κάνει σωστό διαχωρισμό, όπως φαίνεται στον παρακάτω πίνακα.

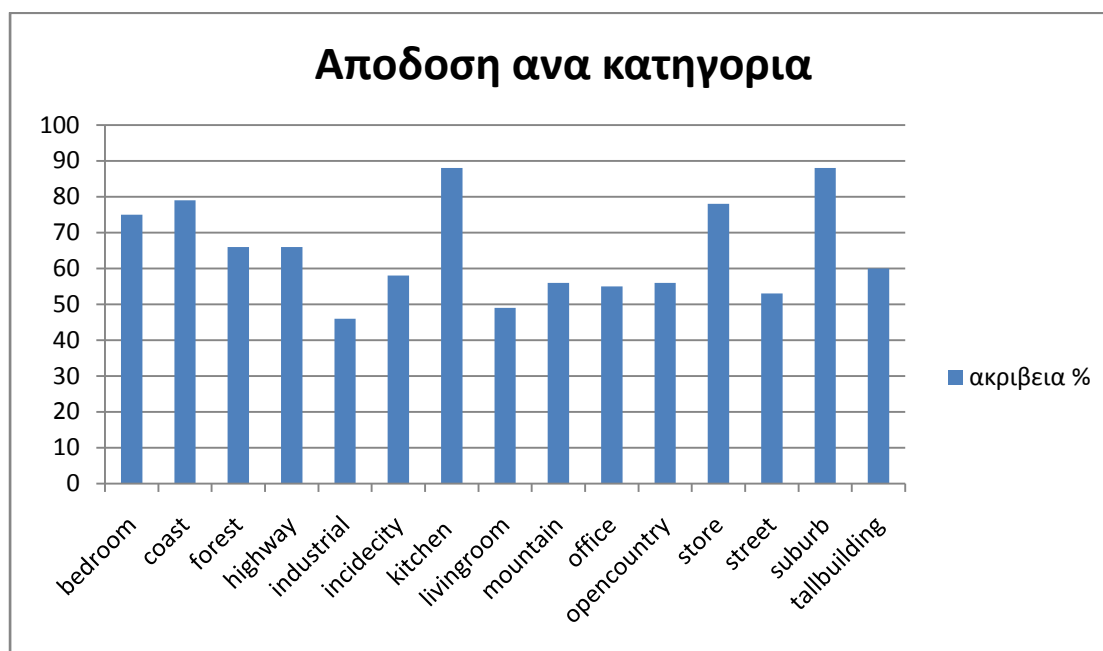
Πίνακας 4.2

Αριθμός κλάσεων	Ποσοστό επιτυχίας %
5	81
10	75
15	65

Επίσης σημαντική βελτίωση στην ακρίβεια του αλγορίθμου παρατηρείται εάν τα δεδομένα που χρησιμοποιούμε είναι κλιμακωτά. Στο συγκεκριμένο παράδειγμα το codebook είναι  $N=400$  και ο αριθμός των σημασιολογικών ετικετών  $C=300$ .

Ένα στοιχείο που είναι άξιο αναφοράς είναι ότι ο αλγόριθμος έχει μεγάλη διακύμανση απόδοσης στην ακρίβεια των κλάσεων, όπως φαίνεται στο παρακάτω γράφημα.

Πίνακας 4.3



Όπως φαίνεται ο αλγόριθμος μπορεί να αναγνωρίσει και να ταξινομήσει σωστά κάποιες κλάσεις έχοντας υψηλή απόδοση, όπου το ποσοστό επιτυχίας φτάνει μέχρι το 88%, ενώ για κάποιες άλλες αδυνατεί να κάνει σωστό ταίριασμα.

Για τα πειράματα χρησιμοποιήθηκαν 200 εικόνες για κάθε κατηγορία, 100 για εκπαίδευση και 100 για ταξινόμηση.

Η διαίρεση των εικόνων γίνεται μέχρι το επίπεδο 3(level 2).

Η ποσοστό συμμετοχής των μεθόδων SPM και SM είναι ίσο(συντελεστής  $\alpha=0,5$ ).

Η συνάρτηση πυρήνα που χρησιμοποιήθηκε στις μηχανές διανυσμάτων υποστήριξης είναι γραμμικού πυρήνα.

Στη συγκεκριμένη εργασία χρησιμοποιήθηκε η βάση δεδομένων εικόνων 15scenes dataset.

Για την εξαγωγή των χαρακτηριστικών SIFT και για την κατηγοριοποίηση κ-μέσων χρησιμοποιήθηκε η βιβλιοθήκη VLFeat version 0.9.18 από τον ιστότοπο <http://www.vlfeat.org/index.html>.

Για την ταξινόμηση με μηχανές διανυσμάτων ταξινόμησης χρησιμοποιήθηκε η βιβλιοθήκη libsvm-3.19 από τον ιστότοπο <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

## ΠΑΡΑΡΤΗΜΑ Α

### 1 Εισαγωγή εικόνων

Αρχικά φορτώνουμε τις εικόνες με την συνάρτηση:

```
[images]=images_load (path, extension)
```

Η συνάρτηση παίρνει στην είσοδο το ακριβές μονοπάτι του φακέλου όπου βρίσκονται τα αρχεία και την κατάληξη των αρχείων, και επιστρέφει μία δομή(struct) με ένα πεδίο, κάθε εγγραφή του πεδίου είναι μία εικόνα.

```
function [images]=images_load (path, extension)
fullpath= ([path]);
ext=([extension]);
D=dir([fullpath ext]);
for i=1: length (D)
images(i).img=imread([fullpath D(i).name]);
images (i).img=single (images(i).img);
end
```

Χρησιμοποιούμε την συνάρτηση images\_load για κάθε κλάση ξεχωριστά και μετά ενώνουμε όλες τις κλάσεις μέσα σε μία συνάρτηση με όνομα data\_load. Η data\_load() δεν περιέχει παραμέτρους εισόδου και επιστρέφει το σύνολο των δεδομένων.

```
function data=data_load()

bedroom=images_load('D:\bedroom\','*jpg');
coast=images_load('D:\coast\','*jpg');
forest=images_load('D:\forest\','*jpg');
highway=images_load('D:\highway\','*jpg');
industrial=images_load('D:\industrial\','*jpg');
insidecity=images_load('D:\insidecity\','*jpg');
kitchen=images_load('D:\kitchen\','*jpg');
livingroom=images_load('D:\livingroom\','*jpg');
```

## ΧΩΡΙΚΗ-ΣΗΜΑΣΙΟΛΟΓΙΚΗ ΑΝΤΙΣΤΟΙΧΙΣΗ

```
mountain=images_load('D:\mountain\','*jpg');  
office=images_load('D:\office\','*jpg');  
opencountry=images_load('D:\opencountry\','*jpg');  
store=images_load('D:\store\','*jpg');  
street=images_load('D:\street\','*jpg');  
suburb=images_load('D:\suburb\','*jpg');  
tallbuilding=images_load('D:\tallbuilding\','*jpg');
```

```
data=[bedroom coast forest highway industrial insidacity  
kitchen livingroom mountain office opencountry store  
street suburb tallbuilding];
```

Η συνάρτηση διαβάζει το σύνολο των εικόνων από το σετ εικόνων 15scene dataset.



## 2 Υπολογισμός του codebook

Η συνάρτηση για την εύρεση του λεξιλογίου(codebook) είναι η:

```
function [codebook]=find_codebook(data,codebooklength)
```

Η συνάρτηση παίρνει ως είσοδο το σύνολο των εικόνων και το μέγεθος του codebook και επιστρέφει το codebook. Κάθε λέξη είναι ένα διάνυσμα μήκους 128 που συμβολίζει έναν αντιπροσωπευτικό περιγραφέα.

```
function [cb]=find_codebook(data,codebooklength)

T=[];
for k=1:length(data)           %read images
    I=data(k).img;
    if ndims(I)==3             %check grayscale
        I=single(rgb2gray(I));
    end
    [X, Y]=size(I);
    for l=0:k                  %l=segmentation level
        m=floor((X./(2.^l)));
        n=floor((Y./(2.^l)));
        X=X-(2.^l);
        Y=Y-(2.^l);
        for i=1:m:X
            for j=1:n:Y
                G=I(i:i+(m-1),j:j+(n-1));
                %extract          SIFT          feature
                [~,descriptors]=vl_sift(G);
                descriptors=single(descriptors);
                T=[T descriptors];
            end
        end
    end
end

[cb]=vl_kmeans(T,codebooklength); %find codebook
cb=cb';
```

Η συνάρτηση διαβάζει τις εικόνες μία-μία, ελέγχει αν έχουμε έγχρωμες εικόνες, αν υπάρχουν τις μετατρέπει σε απόχρωση του γκρι, ο αλγόριθμος λειτουργεί μόνο για εικόνες αποχρώσεως του γκρι. Για κάθε εικόνα και για όλες της υποπεριοχές παράγει χαρακτηριστικά SIFT, τα οποία τοποθετούνται όλα σε ένα χώρο (T) . Ο χώρος (T) είναι είσοδος σε k-means για k=N (αριθμός λέξεων). Η έξοδος του k-Means είναι το λεξιλόγιο(codebook==N).

### 3 Εύρεση σημασιολογικών ετικετών

Η συνάρτηση καλείται για να δημιουργηθούν οι σημασιολογικές ετικέτες είναι:

```
function [semantic_label]=produce_semantic_label(data,
cd, codebooklength, labelsnum)
```

Η συνάρτηση παίρνει σαν είσοδο το σύνολο των εικόνων, το λεξιλόγιο(codebook) και τον αριθμό των ετικετών και έχει σαν έξοδο M διανύσματα χαρακτηριστικών που το καθένα αντιπροσωπεύει μία σημασιολογική ετικέτα.

```
function [semantic_label]=produce_semantic_label_cb(data,
cd, codebooklength, labelnum)
```

```
S=[];
for k=1:length(data)
    I=data(k).img;
    if ndims(I)==3
        I=rgb2gray(I);
    end
    I=single(I);
    [X, Y]=size(I);
    for l=0:k
        m=floor((X./(2.^l)));
        n=floor((Y./(2.^l)));
        X=X-(2.^l);
        Y=Y-(2.^l);
    for i=1:m:X
        for j=1:n:Y
```

```

G=I(i:i+(m-1),j:j+(n-1));
[~,descriptors]=vl_sift(G);
descriptors=single(descriptors)';
[ed]=euclidean(cd, reg_descr);
%find nearest codeword for every feature
[~,position]=min(ed);
%1*N vector of every subregion
N=hist(position,codebooklength);
S=[S;N];

end

end

end

end

S=S'

%product semantic labels
[semantic_label]=vl_kmeans(S,labelsnum); %m=semantic
labels
semantic_label=semantic_label';

```

Η συνάρτηση διαβάζει όλες τις εικόνες. Για κάθε εικόνα ή υποπεριοχή εξάγει τα χαρακτηριστικά SIFT. Μετριέται η Ευκλείδεια Απόσταση όλων των χαρακτηριστικών της κάθε εικόνας με όλες τις οπτικές λέξεις(codeword) του λεξιλογίου(codebook). Υπολογίζεται για κάθε χαρακτηριστικό σε ποιο από τα N κέντρα βρίσκεται πιο κοντά. Σύμφωνα με αυτή την αντιστοιχία δημιουργείται ένα διάνυσμα με μήκος N για την αναπαράσταση κάθε περιοχής. Όταν υπολογιστούν όλα τα διανύσματα χαρακτηριστικών όλων των εικόνων και των υποπεριοχών τους μπαίνουν σαν είσοδο στον k-Means και κατηγοριοποιούνται σε m σημασιολογικές ετικέτες.

### 3.1 Ευκλείδεια Απόσταση

Για τον υπολογισμό της Ευκλείδειας Απόστασης χρησιμοποιείται η συνάρτηση:

```
function [Ed]=euclidean(cd, reg_descr)
```



Η συνάρτηση παίρνει στην είσοδο το οπτικό λεξιλόγιο(codebook) και τους περιγραφείς μιας περιοχής και επιστρέφει ένα πίνακα με διαστάσεις [αριθμός codebook\*αριθμός χαρακτηριστικών].

```
function [E]=euclidean(cd, reg_descr)
E=zeros(size(cd,1),size(reg_descr,1));
for i=1:size(cd,1)
    for j=1:size(reg_descr,1)
        E(i,j)=sqrt(sum((cd(i,:)-reg_descr(j,:)).^2));
    end
end
```

Η έξοδος είναι η Ευκλείδεια Απόσταση του κάθε χαρακτηριστικού με κάθε κέντρο. Η Ευκλείδεια Απόσταση δίνεται από τον τύπο:

$$\sqrt{\sum(y_i-x_i)^2}$$

## 4 SSM Αναπαράσταση

### 4.1 SPM-SM Αναπαράσταση

Η συνάρτηση παράγει τα τελικά διανύσματα για τις αναπαραστάσεις των εικόνων σε μορφή SPM και SM.

```
function [SPM, SM]=SPM_SM_representation(semantic_label,
codebook, data, codebooklength)
```

Η συνάρτηση παίρνει σαν είσοδο τις σημασιολογικές ετικέτες, το οπτικό λεξιλόγιο(codebook) και το σετ των εικόνων και επιστρέφει το σετ των εικόνων στη τελική τους μορφή

```
function [SPM, SM]=SPM_SM_representation(semantic_label,
codebook, data, codebooklength)
```

```
SPM=[];
```

```
SM=[];
```

```

for k=1:length(data)
    I=data(k).img;
    if ndims(I)==3
        I=rgb2gray(I);
    end

I=single(I);
[X, Y]=size(I);
reg=[];

for l=0:k
    m=floor((X./(2.^l)));
    n=floor((Y./(2.^l)));
    X=X-(2.^l);
    Y=Y-(2.^l);
    for i=1:m:X
        for j=1:n:Y
            G=I(i:i+(m-1),j:j+(n-1));
            [~,descriptors]=vl_sift(G);
            descriptors=single(descriptors);
            reg_descr=descriptors';
            [ed]=euclidean(codebook, reg_descr);
            [~,each_reg_hist]=min(ed);
            N=hist(each_reg_hist, codebooklength);
            reg=[reg;N];
        end
    end
end
end

ED=euclidean(semantic_label, reg);
[~,label]=min(ED);
imag_repr=image_representation(semantic_label, reg,
label);

```

```

SM=[SM;imag_repr];
reg=matrix2line(reg);
SPM=[SPM;reg];
end

```

## 4.2 SPM Αναπαράσταση

Η συνάρτηση διαβάζει όλες τις εικόνες και τις υποπεριοχές τους, παράγει τα διανύσματα χαρακτηριστικών για κάθε περιοχή. Τα διανύσματα αποθηκεύονται σε ένα δισδιάστατο πίνακα όπου κάθε γραμμή του πίνακα είναι ένα διάνυσμα χαρακτηριστικών μιας περιοχής, μετατρέποντας αυτόν το πίνακα σε πίνακα σειρά για κάθε εικόνα παράγεται η SPM αναπαράσταση.

## 4.3 SM Αναπαράσταση

Η συνάρτηση διαβάζει όλες τις εικόνες και τις υποπεριοχές τους, παράγει τα διανύσματα χαρακτηριστικών για κάθε περιοχή. Για κάθε διάνυσμα μιας εικόνας μετρίεται η Ευκλείδεια Απόσταση με κάθε σημασιολογική ετικέτα, το κάθε διάνυσμα παίρνει μία από τις  $m$  σημασιολογικές ετικέτες, τα διανύσματα χαρακτηριστικών που έχουν ίδια ετικέτα προστίθενται μεταξύ τους. Για την δημιουργία του τελικού διανύσματος ευθυγραμμίζονται όλα τα διανύσματα ( $s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_m$ ), όπου  $s_i$  το διάνυσμα με  $i$  ετικέτα, αν κάποια ετικέτα δεν έχει αντιστοιχηθεί σε κανένα διάνυσμα χαρακτηριστικών στη θέση της στο τελικό διάνυσμα τοποθετούνται μηδενικά.

### 4.3.1 Τελικό διάνυσμα SM

Η συνάρτηση παράγει το τελικό διάνυσμα χαρακτηριστικών σε μορφή SM.

```

function [imag_repr]=image_representation(semantic_label,
reg, label)

```

Η συνάρτηση παίρνει είσοδο τις σημασιολογικές ετικέτες, τα ιστογράμματα χαρακτηριστικών μιας εικόνας και την ετικέτα που έχει πάρει το κάθε ιστόγραμμα και παράγει το τελικό διάνυσμα.

```

function [imag_repr]=image_representation(semantic_label,

```

```

reg, label)
S=zeros(size(semantic_label,1),size(semantic_label,2));

for i=1:size(reg,1)
    for j=1:size(semantic_label,1)
        if label(i)==j
            S(j,:)=S(j,:)+reg(i,:);
        end
    end
end
imag_repr=matrix2line(S);

```

Η συνάρτηση δημιουργεί ένα μηδενικό πίνακα  $S$  με σειρές όσες είναι οι σημασιολογικές ετικέτες και στήλες το μήκος των διανυσμάτων χαρακτηριστικών. Έπειτα όποιο διάνυσμα χαρακτηριστικών περιοχής έχει  $i$  ετικέτα προστίθεται στην  $i$  σειρά του πίνακα  $S$ . Τέλος για να παραχθεί το τελικό διάνυσμα οι γραμμές του πίνακα  $S$  ευθυγραμμίζονται.

### 4.3.2 Ευθυγράμμιση εικόνας SM

Για την ευθυγράμμιση καλείται η συνάρτηση

```
function [A]=matrix2line(I)
```

Η συνάρτηση απλά παίρνει ένα δισδιάστατο πίνακα και παράγει ένα πίνακα γραμμή βάζοντας σε σειρά τις γραμμές του πίνακα.

```

function [A]=matrix2line(I)
[X Y]=size(I);
A=[];
for i=1:X
    A=[A I(i,:)];
end

```

## 4.4 Ένωση SPM και SM

Η συνάρτηση παράγει τα τελικά δεδομένα προς ταξινόμηση και προς εκπαίδευση.

```
function [trainlabel, testlabel, SSMtrain, SSMtest]=SSM(Group, SPM, SM, a)
```

Η συνάρτηση παίρνει στη είσοδο τις κατηγορίες των εικόνων, τις αναπαραστάσεις που προέκυψαν από τις μεθόδους SPM και SM και τον συντελεστή  $a$  και τα χωρίζει σε πίνακες για εκπαίδευση και ταξινόμηση, υπολογίζει τον πίνακα πυρήνα για κάθε πίνακα (εσωτερικό γινόμενο). Η πρώτη στήλη του πίνακα πυρήνα για εκπαίδευση θα πρέπει να απαριθμεί τις παρατηρήσεις και ο υπόλοιπος πίνακας σχηματίζεται αποκλειστικά την παρακάτω φόρμουλα:

$$0:i \quad 1:K(x_i, x_1) \quad \dots \quad L:K(x_i, x_L)$$

Όπου  $x_1 \dots x_L$  ο αριθμός των παρατηρήσεων  $x_i$  η τρέχουσα παρατήρηση και  $K(x,y)$  η τιμή του πίνακα πυρήνα (εσωτερικό γινόμενο στην προκειμένη περίπτωση που χρησιμοποιείται γραμμικός πυρήνας). Για παράδειγμα ο πίνακας πυρήνας που προκύπτει για τον παρακάτω πίνακα χρησιμοποιώντας γραμμικό πυρήνα θα είναι

Πίνακας:

1	1	1	1
0	3	0	3
0	0	1	0

Πίνακας Πυρήνα:

1	4	6	1
2	6	18	0
3	1	0	1

Για να παραχθεί ο πίνακας πυρήνα για ταξινόμηση χρησιμοποιείται η ίδια φόρμουλα με την προϋπόθεση ότι συμμετέχουν και τα δεδομένα εκπαίδευσης. Για παράδειγμα για το την παρακάτω παρατήρηση θα έχουμε

Παρατήρηση:

1	0	1	0
---	---	---	---

Πίνακας πυρήνα:

$$i \quad 2 \quad 0 \quad 1$$

όπου  $i$  οποιαδήποτε τιμή.

Έπειτα γίνεται ο γραμμικός συνδυασμός των πινάκων πυρήνα των δύο μεθόδων, ώστε να παραχθεί η μέθοδος SSM(χωρική-σημασιολογική ταξινόμηση). Τέλος τα δεδομένα πρέπει να είναι κλιμακωτά για να πετύχουμε μεγαλύτερη ακρίβεια στην ταξινόμηση.

```
function [trainlabel, testlabel, SSMtrain,
SSMtest]=SSM(Group, SPM, SM, a)

[train, test]=crossvalind('HoldOut',Group); %split data
to train and test

SPMtrain=SPM(train,:);
SMtrain=SM(train,:); %train data
trainlabel=Group(train,1);

SPMtest=SPM(test,:);
SMtest=SM(test,:); %testdata
testlabel=Group(test,1);

SPMtrain=kernel(SPMtrain); % kernel matrix
SMtrain=kernel(SMtrain);
SPMtest=kernel(SPMtest);
SMtest=kernel(SMtest);

SSMtrain=a*SPMtrain+(1-a)*SMtrain; % linear combination
SSMtest=a*SPMtest+(1-a)*SMtest;

SSMtrain=zscore(SSMtrain); %scale feature
SSMtest=zscore(SSMtest);
```

#### 4.4.1 Υπολογισμός πίνακα πυρήνα

Η συνάρτηση υπολογίζει τον γραμμικό πίνακα πυρήνα ενός πίνακα.

```
function S=kernel(data1, data2)
```

Η συνάρτηση παίρνει στην είσοδο ένα πίνακα και βρίσκοντας το εσωτερικό γινόμενο των στοιχείων παράγει τον πίνακα πυρήνα.

```
function S=kernel(data1,data2)

S=zeros(size(data1,1),size(data1,1));

for i=1:size(data1,1)
    for j=1:size(data1,1)
        S(i,j)=dot(data1(i,:),data2(j,:));
    end
end
```

#### 4.4.2 Υπολογισμός βέλτιστων παραμέτρων

Η συνάρτηση παράγει τη βέλτιστη τιμή για τη παράμετρο C.

```
function [best_C]=findbestc(trainlabel, SSMtrain)

[~,~,labels]=unique(trainlabel);
%# grid of parameters
folds = 5;
[C] = meshgrid(-5:2:15);

%#grid search, and cross-validation

cv_acc = zeros(numel(C),1);
for i=1:numel(C)
    cv_acc(i) = svmtrain(labels, SSMtrain, ...
        sprintf('-c %f -v %d', 2^C(i),
folds));
end

%# pair (C,gamma) with best accuracy
[~,idx] = max(cv_acc);

%# now you can train you model using best_C and
best_gamma
best_C = 2^C(idx);
```

#### 4.5 Ορισμός κατηγοριών

Για να ορίσουμε τη κατηγορία της κάθε εικόνας καλούμε τη συνάρτηση

```
function Group=categories();
```

Η συνάρτηση δεν έχει κάποια είσοδο και επιστρέφει σαν έξοδο ένα πίνακα στήλη κελιών που δείχνει τη κλάση που ανήκει η κάθε εικόνα.

```
function Group=categories()  
  
Group=cell(3000,1);  
  
for i=1:200  
    Group(i,1)=cellstr('bedroom');  
end  
for i=201:400  
    Group(i,1)=cellstr('coast');  
end  
for i=401:600  
    Group(i,1)=cellstr('forest');  
end  
for i=601:800  
    Group(i,1)=cellstr('highway');  
end  
for i=801:1000  
    Group(i,1)=cellstr('industrial');  
end  
for i=1001:1200  
    Group(i,1)=cellstr('insidecity');  
end  
for i=1201:1400  
    Group(i,1)=cellstr('kitchen');  
end  
for i=1401:1600  
    Group(i,1)=cellstr('livingroom');  
end  
for i=1601:1800  
    Group(i,1)=cellstr('mountain');  
end  
for i=1801:2000  
    Group(i,1)=cellstr('office');  
end  
for i=2001:2200  
    Group(i,1)=cellstr('opencountry');  
end  
for i=2201:2400  
    Group(i,1)=cellstr('store');  
end  
for i=2401:2600  
    Group(i,1)=cellstr('street');  
end  
for i=2601:2800
```



```

    Group(i,1)=cellstr('suburb');
end
for i=2801:3000
    Group(i,1)=cellstr('tallbuilding');
end

```

#### 4.6 Ταξινόμηση με μηχανές διανυσμάτων υποστήριξης

Για την ταξινόμηση των εικόνων καλείται η συνάρτηση

```

function [acc, C]=multisvm(trainlabel, traindata, bestc,
testdata)

```

Η συνάρτηση δέχεται στην είσοδο τα δεδομένα προς εκπαίδευση με τις αντίστοιχες κλάσεις, τα δεδομένα προς κατηγοριοποίηση με τις αντίστοιχες κλάσεις, οι οποίες είναι ίδιες με αυτές των δεδομένων εκπαίδευσης, και τέλος τη παράμετρο C.

```

function [acc, C]=multisvm(trainlabel, traindata, bestc,
testdata)

[~,~,labels]=unique(trainlabel);
numLabels=max(labels);

model = cell(numLabels,1);
for k=1:numLabels
    model{k} = svmtrain(double(labels==k), traindata,
sprintf(' -t 0 -b 1 -c %d', bestc));
end

%# get probability estimates of test instances using each
model
prob = zeros(size(labels,1),numLabels);
for k=1:numLabels
    [~,~,p] = svmpredict(double(labels==k), testdata,
model{k}, '-b 1');
    prob(:,k) = p(:,model{k}.Label==1);    %# probability
of class==k
end

%# predict the class with the highest probability
[~,pred] = max(prob,[],2);
acc = sum(pred == labels) ./ numel(labels);    %#
accuracy

```

```
C = confusionmat(labels, pred); %#
confusion matrix
```

Η συνάρτηση αρχικά εκπαιδεύεται παράγοντας ένα μοντέλο για κάθε δυαδικό ταξινομητή το οποίο χρησιμοποιείται για την πρόβλεψη των δεδομένων προς ταξινόμηση. Τέλος υπολογίζεται η κλάση με την μεγαλύτερη πιθανότητα.

#### 4.5 Οδηγός συναρτήσεων

```
[data]=data_load(); → [images]=images_load (path,
extension);
```

```
[Group]=categories();
```

```
[codebook]=find_codebook(data, codebooklength);
```

```
[semantic_label]=produce_semantic_label(data, codebook,
codebooklength, labelsnum); → [Ed]=euclidean(codedook,
reg_descr);
```

```
[SPM, SM]=SPM_SM_representation(semantic_label, codebook,
data, codebooklength); → [Ed]=euclidean(codedook,
reg_descr); &&
[imag_repr]=image_representation(semantic_label, reg,
label); && A]=matrix2line(I);
```

```
[trainlabel, testlabel, SSMtrain, SSMtest]=SSM(Group,
SPM, SM); → S=kernel(data);
```

```
[bestc, bestg]=findbestc(TrainLabel, TrainVec);
```

```
[acc, C]=multisvm(trainlabel, traindata, bestc,
testdata);
```

## ΠΑΡΑΡΤΗΜΑ Β

### Εφαρμογή

Η συγκεκριμένη εφαρμογή δίνει τη δυνατότητα στο χρήστη να εκπαιδεύσει τον αλγόριθμο με το δικό του σετ εικόνων, και στη συνέχεια επιλέγοντας τυχαίες εικόνες μέσω ενός παράθυρου διαλόγου να προβλέπει την κλάση στην οποία ανήκουν.

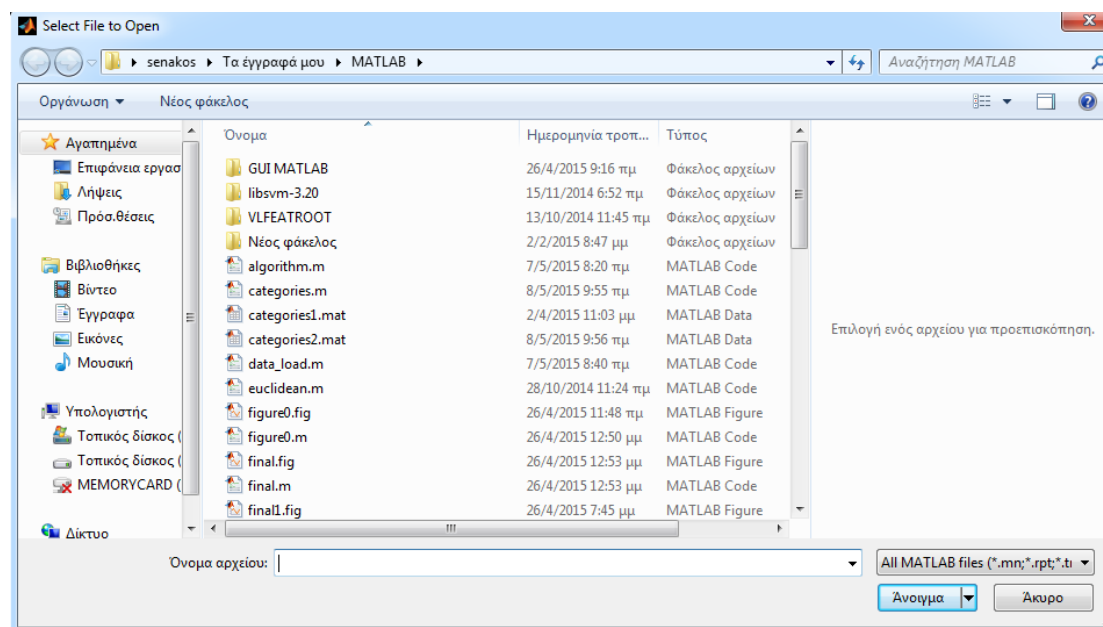
Αρχικά ο χρήστης καλείται να συμπληρώσει δύο EditBox, ένα με το ακριβές μονοπάτι στο οποίο βρίσκεται το σετ των εικόνων (π.χ. D:\GUI\example\ ) και ένα με την κατάληξη των εικόνων (π.χ. \*.jpg), οι εικόνες φορτώνονται πατώντας το κουμπί

Φόρτωσε τις εικόνες

Ο χρήστης θα πρέπει να έχει αποθηκευμένο στο κατάλογο MATLAB ένα αρχείο .mat το οποίο θα περιέχει τις κλάσεις των εικόνων. Πατώντας το πλήκτρο

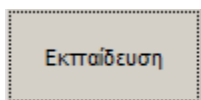
Φόρτωσε τις κλάσεις

ανοίγει ένα παράθυρο διαλόγου

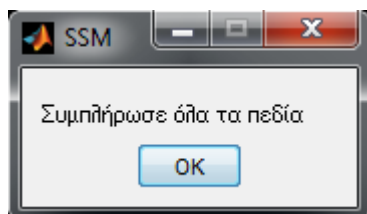


όπου ο χρήστης επιλέγει το .mat αρχείο.

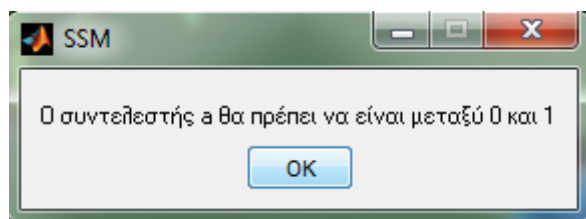
Έπειτα ο χρήστης καλείται να συμπληρώσει τις παραμέτρους για την εκπαίδευση του αλγορίθμου Πατώντας το πλήκτρο



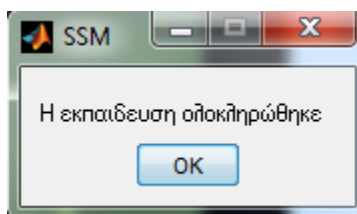
εάν ο χρήστης δεν έχει συμπληρώσει όλα τα πεδία εμφανίζεται μήνυμα λάθους,



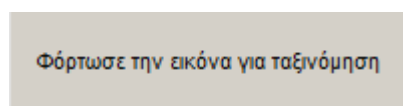
εάν για τον συντελεστής  $\alpha$  έχει δοθεί τιμή έξω από τα όρια  $[0,1]$  εμφανίζεται επίσης ανάλογο μήνυμα λάθους,



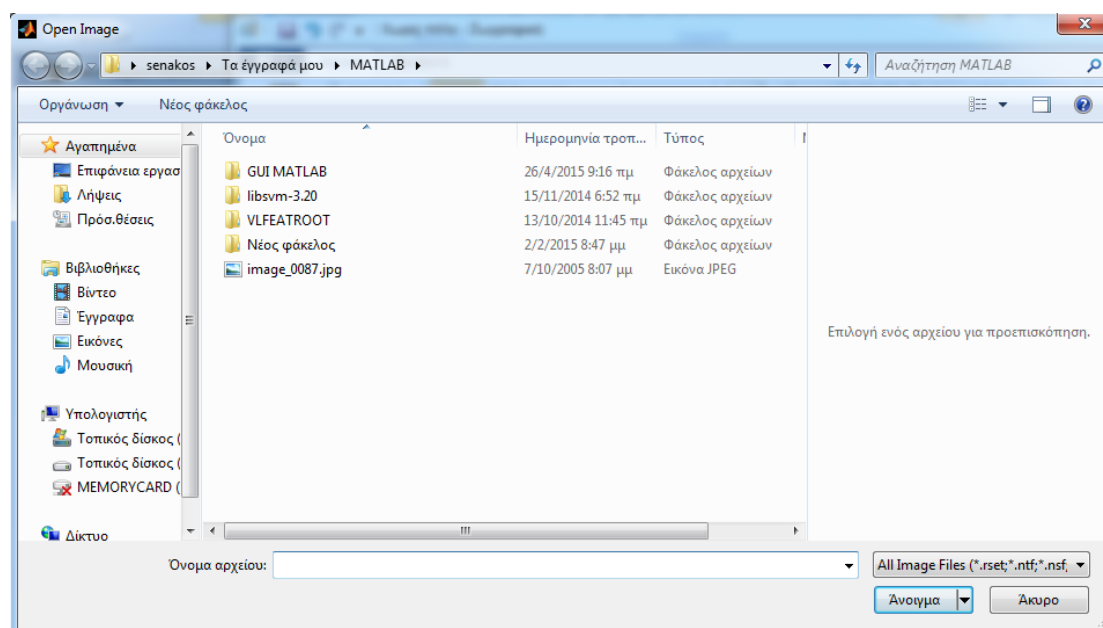
εικότως ο αλγόριθμος εκπαιδεύεται. Όταν ολοκληρωθεί η εκπαίδευση εμφανίζεται ανάλογο μήνυμα στην οθόνη.



Έπειτα πατώντας το πλήκτρο



ανοίγει ένα παράθυρο διαλόγου



από το οποίο ο χρήστης μπορεί να επιλέξει την εικόνα για ταξινόμηση.

Τέλος πατώντας το κουμπί

Κατηγοριοποίηση

Εμφανίζεται η εικόνα που επιλέχθηκε και η κλάση που της απονεμήθηκε.

## ΠΑΡΑΤΗΡΗΣΗ

Για τις ανάγκες της εφαρμογής έχουν γίνει κάποιες μετατροπές στον κώδικα:

1) Η συνάρτηση SSM για την SSM Αναπαράσταση της εικόνας δέχεται ένα σετ εικόνων και το χωρίζει προς εκπαίδευση και προς ταξινόμηση. Στην εφαρμογή όμως χρειάζεται να χρησιμοποιηθούν δύο ξεχωριστές συναρτήσεις, μία για το σετ εικόνων για εκπαίδευση

```
function [SSMtrain]=trainSSM(SPMtrain, SMtrain, a)
```

```
kSPMtrain=kernel(SPMtrain,SPMtrain); % kernel matrix
```

```
kSPMtrain=[(1:size(kSPMtrain,1))',kSPMtrain];
```

```
kSMtrain=kernel(SMtrain,SMtrain);
```

```
kSMtrain=[(1:size(kSMtrain,1))',kSMtrain];
```

## ΧΩΡΙΚΗ-ΣΗΜΑΣΙΟΛΟΓΙΚΗ ΑΝΤΙΣΤΟΙΧΙΣΗ

```
SSMtrain=a*kSPMtrain+(1-a)*kSMtrain; % linear combination
```

```
SSMtrain=zscore(SSMtrain); %scale feature
```

και μια για την εικόνα προς ταξινόμηση

```
function [SSMtest]=testSSM(SPMtest, SMtest, SPMtrain,  
SMtrain, a)
```

```
kSPMtest=kernel(SPMtest,SPMtrain); % kernel matrix
```

```
kSPMtest=[(1:size(kSPMtest,1))',kSPMtest];
```

```
kSMtest=kernel(SMtest,SMtrain);
```

```
kSMtest=[(1:size(kSMtest,1))',kSMtest];
```

```
SSMtest=a*kSPMtest+(1-a)*kSMtest; % linear combination
```

```
SSMtest=zscore(SSMtest); %scale feature
```

2) Η συνάρτηση `multisvm` χωρίζεται σε δύο ξεχωριστές συναρτήσεις όπου η μία χρησιμοποιείται για την εκπαίδευση και παράγει το μοντέλο εκπαίδευσης, και η άλλη χρησιμοποιείται για την ταξινόμηση όπου δίνει την κλάση που έχει προβλεφθεί.

```
function [model]=multisvml(trainlabel, traindata, bestc)
```

```
[~,~,labels]=unique(trainlabel);  
numLabels=max(labels);
```

```
model = cell(numLabels,1);
```

```
for k=1:numLabels
```

```
    model{k} = svmtrain(double(labels==k), traindata,  
    sprintf(' -t 0 -b 1 -c %d', bestc));
```

```
end
```

και

```
function [Tag]=multisvm2(trainlabel, model, testdata)
```

```

[A,~,labels]=unique(trainlabel);
numLabels=max(labels);
prob = zeros(size(testdata,1),numLabels);
for k=1:numLabels
    [~,~,p] = svmpredict(1, testdata, model{k}, '-b 1');
    prob(:,k) = p(:,model{k}.Label==1);    %# probability
of class==k
end

%# predict the class with the highest probability
[~,pred] = max(prob,[],2);
Tag=A(pred,1);

```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," ICCV, pp. 1470–1477, 2003.
- [2] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," CVPR, pp. 1794–1801, 2009.
- [3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," CVPR, 2006.
- [4] G. Csurka, J. Willamowski, C.R. Dance, L. Fan, and C. Bray, "Visual categorization with bags of keypoints," ECCV Workshop on SLCV, pp. 1–22, 2004.
- [5] D.G. LOWE, "Distinctive image features from scaleinvariant keypoints," IJCV, vol. 60, no. 2, pp. 91–110, 2004.
- [6] S. Savarese, J. Winn, and A. Criminisi, "Discriminative object class models of appearance and shape by corelations," CVPR, pp. 2033–2040, 2006.
- [7] X. Li, Y. Song, Y. Lu, and Q. Tian, "Spatial pooling for transformation invariant image representation," ACM Multimedia, pp. 1509–1512, 2011.
- [8] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Spatial-bag-of-features," CVPR, 2010.
- [9] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling features for large-scale partial-duplicate web image search," CVPR, 2009.
- [10] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," BMVC, 2002.
- [11] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li, "Descriptive visual words and visual phrases for image applications," ACM Multimedia, 2009.
- [12] J. Yuan, Y.Wu, and M. Yang, "Discovery of collocation patterns: from visual words to visual phrases," CVPR, pp. 1–8, 2007.
- [13] M. Everingham, L. Van-Gool, C.Williams, J.Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge 2011 (VOC2011) Results," <http://www.pascalnetwork.org/challenges/VOC/voc2011/workshop/index.html>.
- [14] D. Xu, T.-J. Cham, S. Yan, and S.-F. Chang, "Near duplicate image identification with spatially aligned pyramid matching," CVPR, 2008.
- [15] Y. Rubner, C. Tomasi, and L.J. Guibas, "The earth mover's distance as a metric for image retrieval," IJCV, vol. 40, no. 2, pp. 99–121, 2000.
- [16] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zengfu Wang, "Joint multilabel multi-instance learning for image classification," CVPR, pp. 1–8, 2008.
- [17] "Trecvid video retrieval evaluation," <http://www.nlpir.nist.gov/projects/trecvid/>.
- [18] Meng Wang, Xian-Sheng Hua, Richang Hong, Jinhui Tang, Guo-Jun Qi, and Yan Song, "Unified video annotation via multigraph learning," IEEE TCSVT, vol. 19, no. 5, pp. 733–746, 2009.
- [19] Jinhui Tang, Zheng-Jun Zha, Dacheng Tao, and Tat-Seng Chua, "Semantic-gap-oriented active learning for multilabel image annotation," IEEE TIP, vol. 21, no. 4, pp. 2354–2360, 2012.
- [20] Li F.-F. and P. Pietro, "A bayesian hierachical model for learning natural scene categories," CVPR, 2005.
- [21] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," IJCV, vol. 42, no. 3, pp. 145–175, 2001
- [22] SEMANTIC-SPATIAL MATCHING FOR IMAGE CLASSIFICATION.
- [23] <http://www.vlfeat.org/index.html>.
- [24] <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [25] <http://stackoverflow.com/questions/9041753/multi-class-classification-in-libsvm>
- [26] [http://www.mathworks.com/matlabcentral/?s\\_tid=gn\\_mlc\\_logo](http://www.mathworks.com/matlabcentral/?s_tid=gn_mlc_logo)